

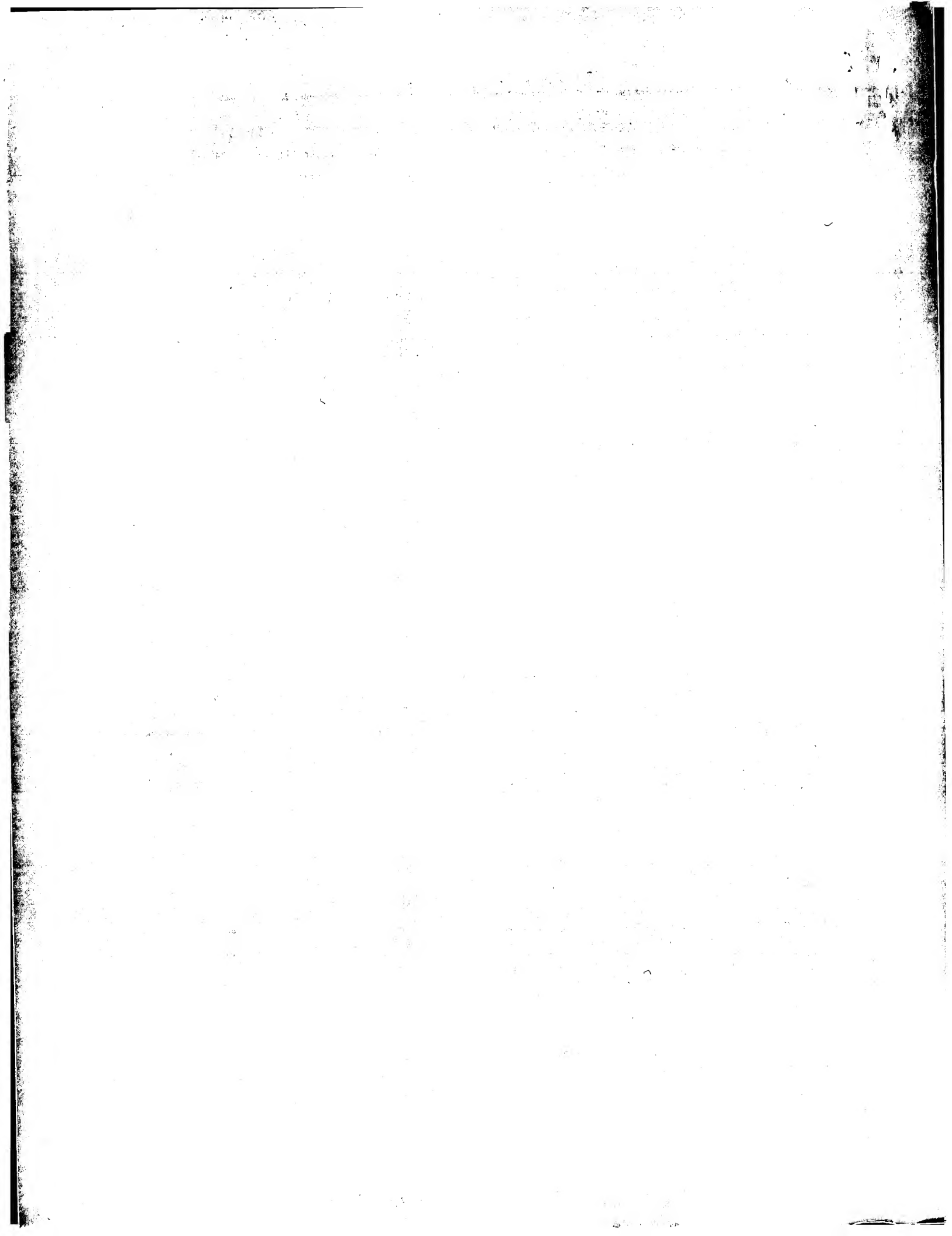
Shadow function box interface for use in process control network e.g. chemical or petroleum process has memory receiving data pertaining to external function block, according to configuration protocol of internal function block

Patent Number: DE19940230
Publication date: 2000-03-16
Inventor(s): CHRISTENSEN DAN D (US); JUNDT LARRY O (US); TZOVLAS VASILIKI (US); DIENSTBIER STEVE L (US); STEVENSON DENNIS L (US)
Applicant(s): FISHER ROSEMOUNT SYSTEMS (US)
Requested Patent: ☐ DE19940230
Application Number: DE19991040230 19990825
Priority Number (s): US19980151084 19980910
IPC Classification: G05B19/042; G06F13/00
EC Classification:
Equivalents: ☐ GB2341524, ☐ JP2000216847

Abstract

A shadow function block (108) has a communications port with an input that communicates with an external function block (112) via the communication network. This is done to receive data pertaining to the external function block. A memory (156) stores the received data pertaining to the external function block, according to a configuration protocol of the internal function block (152). Independent claims are also included for (1) A controller adapted to be communicatively coupled to a number of field devices, and (2) A method of implementing a control routine within a process controller.

Data supplied from the esp@cenet database - I2





①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ Off nlegungsschrift
⑩ DE 199 40 230 A 1

⑤1 Int. Cl.⁷:
G 05 B 19/042
G 06 F 13/00

②1 Aktenzeichen: 199 40 230.2
②2 Anmeldetag: 25. 8. 1999
④3 Offenlegungstag: 16. 3. 2000

DE 199 40 230 A 1

- ③0 Unionspriorität:
151084 10. 09. 1998 US
- ⑦1 Anmelder:
Fisher-Rosemount Systems., Inc.
(n.d.Ges.d.Staates Delaware), Austin, Tex., US
- ⑦4 Vertreter:
Meissner, Bolte & Partner, 80538 München

- ⑦2 Erfinder:
Stevenson, Dennis L., Round Rock, Tex., US; Tzovla,
Vasiliki, Austin, Tex., US; Jundt, Larry O., Round
Rock, Tex., US; Christensen, Dan D., Austin, Tex.,
US; Dienstbier, Steve L., Round Rock, Tex., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

- ⑤4 Interface in Form eines Schattenfunktionsblockes für die Verwendung in einem Prozessregelnetzwerk
- ⑤7 Ein Prozeßkontroller, der kommunikativ mit einer externen Bereichsvorrichtung über ein Kommunikationsnetzwerk gekoppelt ist, verwendet einen Schattenfunktionsblock, der innerhalb eines Prozeßkontrollers angeordnet ist, um eine Implementierung einer Steuer- oder Routineweise zu ermöglichen, die sowohl einen internen Funktionsblock, der innerhalb des Prozeßkontrollers angeordnet ist, als auch einen externen Funktionsblock, der innerhalb der externen Bereichsvorrichtung angeordnet ist, verwendet. Der Schattenfunktionsblock enthält einen Kommunikationsport, der mit dem externen Funktionsblock über das Kommunikationsnetzwerk kommuniziert, um dadurch Daten, die den externen Funktionsblock betreffen, zu empfangen, enthält einen Speicher, der die empfangenen Daten gemäß einem Konfigurationsprotokoll des internen Funktionsblocks speichert, und einen Ausgang, der die gespeicherten externen Funktionsblockdaten in den internen Funktionsblock gemäß dem Konfigurationsprotokoll des internen Funktionsblocks liefert. Der Kommunikationsport des Interface-Funktionsblocks kann einen Ausgang enthalten, der Daten, die durch den Kontroller oder den internen Funktionsblock erzeugt worden sind, zu dem externen Funktionsblock unter Verwendung des Kommunikationsprotokolls sendet, welches dem externen Funktionsblock zugeordnet ist.

DE 199 40 230 A 1

Die vorliegende Erfindung betrifft Prozeßregelnetzwerke und spezieller einen Prozeßkontroller, der Schattenfunktionsblöcke verwendet, um Informationen wiederzugeben, die externen Funktionsblöcken zugeordnet sind, welche in einem Prozeßregelnetzwerk verteilt sind.

Prozeßregelnetzwerke, wie beispielsweise solche, die in chemischen, Mineralöl- oder anderen Prozessen verwendet werden, enthalten im allgemeinen einen zentralisierten Prozeßkontroller, der kommunikativ mit einer oder mehreren Gebiets- oder Bereichsvorrichtungen gekoppelt ist, die beispielsweise Ventilstellglieder, Schalter, Sensoren (wie z. B. Temperatur-, Druck- und Strömungsratesensoren) usw. sein können. Diese Bereichsvorrichtungen können Steuerfunktionen innerhalb des Prozesses durchführen (wie das Öffnen oder Schließen eines Ventils), können Messungen innerhalb des Prozesses vornehmen, die für die Steuerung oder Regelung der Arbeitsweise des Prozesses verwendet werden, oder können irgendeine andere gewünschte Funktion innerhalb des Prozesses ausführen. Prozeßregler wurden in historischer Weise an die Bereichsvorrichtungen über eine oder mehrere analoge Signalleitungen oder Busse angeschlossen, die beispielsweise 4–20 mA (Milliampere) Signale zu und von den Bereichsvorrichtungen leiten. Allgemein gesagt, empfängt der Prozeßkontroller Signale, welche die Messungen angeben, die durch einen oder durch mehrere Bereichsvorrichtungen vorgenommen wurden und/oder andere Informationen anzeigen, die zu einer oder zu mehreren Bereichsvorrichtungen gehören, und er verwendet diese Informationen, um eine typische komplexe Steuerroutine zu implementieren, und erzeugt dann Steuersignale, die über die analogen Signalbusse zu einer oder mehreren der Bereichsvorrichtungen gesendet werden, um dadurch den Betrieb des Prozesses zu steuern.

Kürzlich kam Bewegung in die Prozeßregel- oder -steuerindustrie, um bereichsbasierende digitale Kommunikationen in der Prozeßregelumgebung zu implementieren. Beispielsweise hat die Prozeßregelindustrie eine Anzahl von offenen, digitalen oder kombinierten digitalen und analogen Standardkommunikationsprotokollen entwickelt, wie beispielsweise die HART®, PROFIBUS®, WORLDVIEW®, Device-Net® und CAN-Protokolle. Diese digitalen Kommunikationsprotokolle ermöglichen es mehreren Bereichsvorrichtungen, an einen bestimmten Bus angeschlossen zu werden, sie unterstützen mehrere und schnelle Kommunikationen zwischen den Bereichsvorrichtungen und dem Kontroller und/oder ermöglichen es den Bereichsvorrichtungen, mehrere und unterschiedliche Typen von Informationen, wie beispielsweise Informationen, die den Status und die Konfiguration der Bereichsvorrichtung selbst betreffen, zu dem Prozeßkontroller zu senden. Ferner ermöglichen es diese digitalen Standardprotokolle den Bereichsvorrichtungen, die von unterschiedlichen Herstellern hergestellt wurden, zusammen mit dem gleichen Prozeßregelnetzwerk verwendet zu werden.

Auch gibt es nunmehr Bewegung innerhalb der Prozeßregelindustrie, die Prozeßregelung oder -steuerung zu dezentralisieren, wodurch die Prozeßkontroller vereinfacht werden. Eine dezentralisierte Steuerung oder Regelung wird dadurch erhalten, daß bereichsmontierte Prozeßsteuervorrichtungen, wie beispielsweise Ventilstellglieder, Sender usw., eine oder mehrere Prozeßsteuerfunktionen durchführen, und zwar unter Verwendung von Einrichtungen, die in typischer Weise als Funktionsblöcke bezeichnet werden, und indem dann Daten über eine Busstruktur für die Verwendung durch andere Prozeßsteuervorrichtungen (oder Funktionsblöcke) bei der Ausführung anderer Steuerfunktionen übertragen

werden. Um diese Steuerfunktionen zu implementieren, enthält jede Prozeßsteuervorrichtung in typischer Weise einen Mikroprozessor mit der Fähigkeit, einen oder mehrere Funktionsblöcke zu implementieren, als auch die Fähigkeit, mit anderen Prozeßsteuervorrichtungen zu kommunizieren, und zwar unter Verwendung eines offenen Standardkommunikationsprotokolls. In dieser Weise können Bereichsvorrichtungen innerhalb eines Prozeßregelnetzwerks miteinander verbunden werden, um miteinander zu kommunizieren und um eine oder mehrere Prozeßsteuerfunktionen unter Bildung einer Regelschleife durchzuführen, und zwar ohne Einschreiten eines zentralisierten Prozeßkontrollers. Das Gesamtdigitale Zweidraht-Busprotokoll, welches durch die Fieldbus Foundation entworfen wird, bekannt als FOUNDATION (eingetragene Marke) Fieldbus- (im folgenden als "Fieldbus" bezeichnet) Protokoll bekannt ist, ist ein offenes Kommunikationsprotokoll, welches es den Vorrichtungen, die von unterschiedlichen Herstellern stammen, ermöglicht, untereinander zusammenzuarbeiten und zu kommunizieren, und zwar über einen Standardbus, um eine dezentralisierte Steuerung innerhalb des Prozesses zu bewirken.

Da digitale Kommunikationsprotokolle und dezentralisierte Steuerungsschemata (wie solche, die in der Fieldbus-Steuer- oder -Regelumgebung verwendet werden) so neu sind, verwenden Prozesse, welche diese Protokolle implementieren, diese lediglich in einem eingeschränkten Ausmaß. Als ein Ergebnis unterstützen neuere Prozeßkontroller, wie der Delta V- (eingetragene Marke) Prozeßkontroller, der durch Fisher-Rosemount Systems hergestellt wird, sowohl analoge als auch digitale Kommunikationsprotokolle und es kann eine Hardware so programmiert werden, um die Steuerung in einem Prozeß zu implementieren, der Bereichsvorrichtungen enthält, die unter Verwendung von analogen Standardprotokollen kommunizieren, wie beispielsweise einem 4–20 mA Protokoll, und unter Verwendung von einem oder mehreren neueren digitalen Protokollen, wie dem Fieldbus-Protokoll.

Es stellten sich jedoch Probleme ein, wenn versucht wurde, die Steuerung von analogen und digitalen Bereichsvorrichtungen zu integrieren, und zwar speziell bei den Fieldbus-Bereichsvorrichtungen in einem Prozeßregelnetzwerk, welches einen zentralisierten Kontroller verwendet. Da die Steuerfunktionen für analoge Bereichsvorrichtungen und einige digitale Bereichsvorrichtungen innerhalb des zentralisierten Prozeßkontrollers implementiert sind, werden alle erforderlichen Informationen über diese Bereichsvorrichtungen innerhalb des zentralisierten Prozeßkontrollers empfangen und gespeichert. Dies macht es seinerseits möglich, daß der zentralisierte Prozeßkontroller die Steuerung unterschiedlicher analoger und digitaler Bereichsvorrichtungen integriert, die die momentane Konfiguration und Zustand von Abschnitten des Prozeßregelnetzwerks darstellt, in welchem diese Vorrichtungen gelegen sind, um Änderungen in der Prozeßregelnetzwerkkonfiguration in bezug auf diese Vorrichtungen usw. vorzunehmen. Wenn jedoch ein dezentralisiertes Steuerschema in einem Teil des Prozesses verwendet wird, braucht der zentralisierte Prozeßkontroller keinen direkten Zugriff mehr auf all die laufenden Informationen zu haben, die durch die Prozeßsteuervorrichtungen verwendet werden oder diesen zugeordnet sind, die der dezentralisierten Steuerung unterworfen sind. Tatsächlich sind einige dezentralisierte Prozeßsteuerprotokolle, wie beispielsweise das Fieldbus-Protokoll, spezifisch so ausgelegt, daß es nicht erforderlich ist, Informationen zu einem zentralisierten Prozeßkontroller auf einer regulären Grundlage zu senden. Diese Tatsache macht es für den zentralisierten Prozeßkontroller schwierig, die Steuerung von analogen oder anderen

digitalen Bereichsvorrichtungen mit den Bereichsvorrichtungen zu integrieren, die der dezentralisierten Steuerung unterworfen sind. Es macht es auch für den zentralisierten Prozeßkontroller schwierig, den laufenden oder momentanen Zustand der Vorrichtungen als Modell nachzubilden oder darzustellen, die der dezentralisierten Steuerung unterworfen sind, oder dies innerhalb eines dezentralisierten Abschnitts des Prozeßregelnetzwerks durchzuführen.

Tatsächlich müssen für den zentralisierten Prozeßkontroller, der die Informationen von dem dezentralisierten Steuerabschnitt des Prozeßregelnetzwerks empfängt, die Bereichsvorrichtungen (oder Funktionsblöcke) innerhalb dieses Abschnitts des Prozesses spezifisch konfiguriert sein, um Informationen direkt zu dem zentralisierten Prozeßkontroller zu senden (was bedeutet, daß der zentralisierte Prozeßkontroller all diese Informationen empfangen und handhaben muß, wobei der größte Teil derselben für die Betriebsweise des zentralisierten Prozeßkontrollers nicht erforderlich ist). Alternativ muß der zentralisierte Prozeßkontroller aktiv anfragen, um die von ihm benötigten Informationen zu empfangen. Da solch einer Anfrage keine höhere Priorität in beispielsweise dem Feldbus-Protokoll gegeben wird, und zwar zu der Zeit, in der der zentralisierte Prozeßkontroller die angefragten Informationen empfängt, können diese Information veraltet sein. Es ist ferner für den zentralisierten Prozeßkontroller schwierig, wenn nicht unmöglich, Daten anzufordern und zu empfangen, die zu einer spezifisierten Zeit oder Zeitpunkt aktiv oder momentan vorhanden sind. Statt dessen empfängt der zentralisierte Prozeßkontroller lediglich die Daten zu dem Zeitpunkt, zu welchem die Anfrage die Bereichsvorrichtung erreicht. Darüber hinaus ist die Kommunikation zwischen dem zentralisierten Prozeßkontroller und den Bereichsvorrichtungen innerhalb des dezentralisierten Abschnitts des Prozesses hoch spezialisiert und erfordert ein beträchtliches Wissen über das dezentralisierte Steuerprotokoll, was es für den Konstrukteur oder Entwickler der zentralisierten Prozeßsteuerroutine schwierig macht, diese Kommunikation auf einer gewünschten Basis zu implementieren.

Die vorliegende Erfindung zielt auf die Verwendung von Schattenfunktionsblöcken ab, um in einem zentralisierten Prozeßkontroller die Steuerung der Funktionsblöcke, die in dem zentralisierten Kontroller vorherrschen, mit solchen zu integrieren, die in einer externen Vorrichtung vorherrschen, wie beispielsweise einer Bereichsvorrichtung. Die vorliegende Erfindung kann auch dafür verwendet werden, einem Kontroller die Möglichkeit zu geben, Vorrichtungen oder Funktionsblöcke zu steuern oder mit diesen zu kommunizieren, die in einem Kommunikationsnetzwerk gelegen sind oder die ein Kommunikationsprotokoll verwenden, welches von demjenigen verschieden ist, welches der Kontroller verwendet. Speziell wird ein Schattenfunktionsblock in dem zentralisierten Kontroller erstellt oder eingerichtet, um die Daten, die einem externen Funktionsblock zugeordnet sind, der in einer externen Vorrichtung, wie beispielsweise einer Bereichsvorrichtung, gelegen ist, zu spiegeln, und zwar in Einklang mit einem dezentralisierten Steuer- und Kommunikationsprotokoll. Die Steuerroutine des zentralisierten Kontrollers kommuniziert mit dem externen Funktionsblock über den Schattenfunktionsblock, so als ob der externe Funktionsblock durch den zentralisierten Kontroller implementiert wäre. Der Schattenfunktionsblock erhält automatisch die momentanen Informationen, die dem externen Funktionsblock zugeordnet sind, und sendet Befehle zu dem externen Funktionsblock unter Verwendung des Protokolls, welches dem externen (z. B. dezentralisierten) Steuerprotokoll zugeordnet ist. Im Falle eines Feldbus-Protokolls wird diese Kommunikation unter Verwendung von sowohl syn-

chronen als auch asynchronen Kommunikationen erreicht. Jedoch kommuniziert der Schattenfunktionsblock mit anderen Funktionsblöcken innerhalb des zentralisierten Kontrollers so, als ob der externe Funktionsblock vollständig durch den zentralisierten Kontroller implementiert wäre.

Unter Verwendung des Schattenfunktionsblockes der vorliegenden Erfindung kann die zentralisierte Prozeßsteuerroutine auf den neuesten Stand gebrachte Informationen um den tatsächlichen Funktionsblock herum in einer Realzeit oder nahezu auf Realzeitbasis empfangen, da diese Informationen in dem Schattenfunktionsblock gespiegelt werden, der immer für die zentralisierte Prozeßsteuerroutine zugänglich ist. In ähnlicher Weise kann die zentralisierte Prozeßsteuerroutine Befehle zu dem tatsächlichen Funktionsblock dadurch senden, indem sie einen Befehl zu dem Schattenfunktionsblock ungeachtet des Typs oder der Lage der Vorrichtung sendet, in welcher der tatsächliche Funktionsblock gelegen ist. Der Schattenfunktionsblock setzt dann diesen Befehl unmittelbar in Verbindung zu dem tatsächlichen Funktionsblock unter Verwendung geeigneter Kommunikationsbefehle, die dem dezentralisierten Prozeßsteuerprotokoll zugeordnet sind. Auf diese Weise braucht die zentralisierte Prozeßsteuerroutine keine signifikante Datenbasissteuerung in bezug auf die Bereichsvorrichtungen durchzuführen, und zwar innerhalb des dezentralisierten Steuerabschnitts des Prozesses, da die Informationen, die sie für die Bereichsvorrichtungen innerhalb des dezentralisierten Abschnitts des Prozeßregelnetzwerks, in dem Schattenfunktionsblock bzw. den Schattenfunktionsblöcken gelegen sind. In ähnlicher Weise braucht die zentralisierte Prozeßsteuerroutine nicht spezifische programmiert zu werden, um mit den komplexen und fordernden Aufgaben der Kommunikation in dem dezentralisierten Prozeßsteuerprotokoll fertig zu werden, da diese Kommunikation automatisch durch den Schattenfunktionsblock durchgeführt wird.

Gemäß einem Aspekt der vorliegenden Erfindung ist ein Interface oder Schattenfunktionsblock dafür ausgebildet, um in einem Prozeßkontroller verwendet zu werden, der kommunikativ mit einer externen Vorrichtung über ein Kommunikationsnetzwerk gekoppelt ist, um dem Prozeßkontroller die Möglichkeit zu geben, eine Steuerroutine unter Verwendung von sowohl dem internen Funktionsblock, der in dem Prozeßkontroller vorherrscht, als auch eines externen Funktionsblockes, der innerhalb der externen Vorrichtung vorhanden ist, zu implementieren. Das Interface oder der Schattenfunktionsblock gemäß diesem Aspekt der Erfindung enthält einen Kommunikationsport mit einem Eingang, der mit dem externen Funktionsblock über das Kommunikationsnetzwerk kommuniziert, um dadurch Daten zu empfangen, die zum externen Funktionsblock gehören, und enthält einen Speicher, der die empfangenen Daten, die zu dem externen Funktionsblock gehören, gemäß einem Konfigurationsprotokoll des internen Funktionsblockes speichert. Wenn es erwünscht ist, kann der Interface-Funktionsblock auch einen Ausgang aufweisen, der die gespeicherten externen Funktionsblockdaten zu dem internen Funktionsblock liefert, und zwar unter Verwendung des Konfigurationsprotokolls des internen Funktionsblockes. In ähnlicher Weise kann der Kommunikationsport des Interface-Funktionsblockes einen Ausgang enthalten, der Daten (wie beispielsweise verkettete Daten oder Befehle) zu dem externen Funktionsblock sendet, und zwar unter Verwendung eines Kommunikationsprotokolls, welches dem externen Funktionsblock zugeordnet ist.

In bevorzugter Weise empfängt der Kommunikationsport des Interface-Funktionsblockes die Daten, die zu dem externen Funktionsblock gehören, unabhängig von der Operation der internen Funktionsblöcke. In ähnlicher Weise kommuni-

ziert bei einer Ausführungsform der externe Funktionsblock über das Kommunikationsnetzwerk unter Verwendung eines ersten Kommunikationsprotokolls, welches verschieden ist von einem zweiten Kommunikationsprotokoll, welches dem internen Funktionsblock zugeordnet ist, und in diesem Fall kommuniziert der Kommunikationsport mit dem externen Funktionsblock unter Verwendung des ersten Kommunikationsprotokolls und der Ausgang des Interface-Funktionsblocks kommuniziert mit dem internen Funktionsblock gemäß dem zweiten Kommunikationsprotokoll.

Wenn es gewünscht wird, kann das erste Kommunikationsprotokoll, welches dem externen Funktionsblock zugeordnet ist, ein Fieldbus-Kommunikationsprotokoll sein. In diesem Fall kann der Kommunikationsport eine Interfacevorrichtung (wie beispielsweise eine Interfacekarte) verwenden, die dafür konfiguriert ist, um mit der externen Vorrichtung unter Verwendung des Fieldbus-Kommunikationsprotokolls zu kommunizieren, um dadurch eine Kommunikation mit dem externen Funktionsblock vorzunehmen. Der Kommunikationsport kann beispielsweise mit dem externen Funktionsblock unter Verwendung synchroner Kommunikationen kommunizieren, wie beispielsweise die Herausgeber-/Teilnehmerbeziehungen des Fieldbus-Kommunikationsprotokolls und/oder kann asynchrone Kommunikationen verwenden, wie z. B. die Betrachtungsobjekte in dem Fieldbus-Kommunikationsprotokoll. Allgemeiner kann die externe Vorrichtung unter Verwendung eines Vorrichtungskommunikationsprotokolls kommunizieren, welches die Kommunikation der logisch verketteten Pakete an Daten unter Verwendung von standardisierten Kommunikationsrufen unterstützt, wie beispielsweise die Betrachtungsobjekte und die Betrachtungswarnungen des Fieldbus-Kommunikationsprotokolls, und es kann der Kommunikationsport mit dem externen Funktionsblock unter Verwendung der standardisierten Kommunikationsrufe kommunizieren. In ähnlicher Weise kann der Kommunikationsport Alarmanzeigen von dem externen Funktionsblock empfangen und kann diese Alarmanzeigen in dem Speicher für die Verwendung durch den Controller speichern.

Gemäß einem anderen Aspekt der vorliegenden Erfindung enthält ein Controller, der dafür geeignet ist, um kommunikativ an eine Vielzahl von Bereichsvorrichtungen gekoppelt zu werden, einen Speicher und eine Steuerroutine enthalten, die in dem Speicher abgespeichert ist und die durch den Prozessor implementiert ist, um die Vielzahl der Bereichsvorrichtungen zu steuern. Die Steuerroutine enthält eine Vielzahl von miteinander verbundenen internen Funktionsblöcken, die unter Verwendung eines Controllerprotokolls konfiguriert sind, so daß sie durch den Controller implementiert werden, und kann einen Interface-Funktionsblock enthalten, der mit einem der miteinander verbundenen internen Funktionsblöcken kommuniziert, unter Verwendung des Controllerprotokolls und der mit einem externen Funktionsblock kommuniziert, der in einer externen Bereichsvorrichtung vorhanden ist, und zwar unter Verwendung eines Bereichsvorrichtungskommunikationsprotokolls. Der Interface-Funktionsblock speichert Daten, die zu dem externen Funktionsblock gehören und von dem externen Funktionsblock für die Verwendung durch die Steuerroutine empfangen werden.

Gemäß einem weiteren Aspekt der vorliegenden Erfindung speichert ein Verfahren zum Implementieren einer Steuerroutine innerhalb eines Prozeßkontrollers, innerhalb des Kontrollers eine Vielzahl von miteinander verbundenen internen Funktionsblöcken, die gemäß einem Controllerprotokoll konfiguriert sind, um einen Teil der Controllerroutine zu implementieren. Das Verfahren erzeugt auch innerhalb des Kontrollers einen Interface-Funktionsblock, der gemäß

dem Controllerprotokoll konfiguriert ist, der jedoch mit einem externen Funktionsblock kommuniziert, der in einer externen Bereichsvorrichtung gelegen ist, und zwar unter Verwendung eines Vorrichtungskommunikationsprotokolls, welches der externen Vorrichtung zugeordnet ist. Das Verfahren erzeugt dann eine Steuerroutine, welche die Vielzahl der miteinander verbundenen internen Funktionsblöcke und den Interface-Funktionsblock verwendet, um den Prozeß zu steuern oder zu regeln und während der Implementierung der Steuerroutine werden Daten gespeichert, die dem externen Funktionsblock zugeordnet sind und von diesem empfangen werden, und zwar in dem Interface-Funktionsblock für die Verwendung durch die Steuerroutine, so als ob der externe Funktionsblock durch den Controller als einer der internen Funktionsblöcke implementiert wäre.

Das Verfahren kann einem Anwender auch erlauben, die Steuerroutine dadurch zu konfigurieren, indem jeder eine Anzahl von Funktionsblöcken, die in der Steuerroutine verwendet werden sollen, spezifiziert wird, die Verbindungen zwischen den spezifizierten Funktionsblöcken, die in der Steuerroutine zu verwenden sind, identifiziert werden und die Stelle oder Ort eines bestimmten spezifizierten Funktionsblockes spezifiziert wird, und zwar als ein interner Funktionsblock, der in dem Controller implementiert ist, oder als ein externer Funktionsblock, der durch eine Bereichsvorrichtung implementiert ist. In dem Fall, bei welchem ein Funktionsblock in einer externen Vorrichtung zu implementieren ist, enthält das Verfahren den Schritt gemäß einer Auswahl einer externen Vorrichtung aus einer Liste oder einem Satz von Vorrichtungen, die in dem System verbunden oder angeschlossen sind.

Fig. 1 ist ein schematisches Blockschaltbild eines beispielhaften Prozeßregelnetzwerks, welches zentralisierte und dezentralisierte Prozeßsteuerabschnitte darin enthält;

Fig. 2 ist ein schematisches Blockschaltbild von drei Standardfunktionsblöcken, die einer oder mehreren Fieldbus-Bereichsvorrichtungen zugeordnet sind;

Fig. 3 ist ein Zeitsteuerschema für einen Makrozyklus eines Segments eines Fieldbus-Busses, der innerhalb des Prozeßregelnetzwerks von Fig. 1 gelegen ist;

Fig. 4 ist ein schematisches Blockschaltbild, welches die Verbindung eines Schattenfunktionsblockes mit anderen Funktionsblöcken herausgreift, die einer zentralisierten Prozeßsteueroutine zugeordnet sind und innerhalb eines zentralisierten Prozeßkontrollers gelegen sind;

Fig. 5 ist ein Blockdiagramm eines Teiles eines Prozeßregelsystems unter Verwendung des Schattenfunktionsblockes der vorliegenden Erfindung;

Fig. 6 ist ein Flußdiagramm, welches die Installation eines Steuermoduls innerhalb des Kontrollers von Fig. 1 veranschaulicht;

Fig. 7 ist ein Flußdiagramm, welches die Installation eines aktiven Verbindungsgliedplans in dem Fieldbus-Abschnitt des Prozeßregelnetzwerks von Fig. 1 veranschaulicht;

Fig. 8 ist ein Flußdiagramm, welches die Installation von Veröffentlichender-Bindegliedern in dem Fieldbus-Abschnitt des Prozeßregelnetzwerks von Fig. 1 veranschaulicht;

Fig. 9 ist ein Flußdiagramm, welches die Installation einer Vorrichtungskonfiguration in einer Vorrichtung innerhalb des Fieldbus-Abschnitts des Prozeßregelnetzwerks von Fig. 1 veranschaulicht;

Fig. 10 ist ein Flußdiagramm, welches die Installation eines Funktionsblockes innerhalb einer Vorrichtung in dem Fieldbus-Abschnitt des Prozeßregelnetzwerks von Fig. 1 veranschaulicht;

Fig. 11 ist ein Flußdiagramm, welches die Betriebsweise des Funktionsblockes und darauf bezogener Komponenten

während des Betriebs eines zentralisierten Steuermoduls veranschaulicht, und zwar unter Verwendung des Schattenfunktionsblockes; und

Fig. 12 ist ein Flußdiagramm, welches die Kommunikation von Daten und von Schreibenfragen von einem Block in einem zentralisierten Kontroller oder einem Anwender zu einem externen Funktionsblock in einer Bereichsvorrichtung unter Verwendung des Schattenfunktionsblockes der Erfindung veranschaulicht.

Gemäß Fig. 1 ist ein Prozeßregelnetzwerk 10 in Blockdiagrammform veranschaulicht. Das Prozeßregelnetzwerk 10 enthält einen zentralisierten Prozeßkontroller 12, der eine Prozeßsteueroutine, die in diesem gespeichert ist, implementieren kann. Der Kontroller 12 kann, um hier lediglich ein Beispiel zu nennen, der DeltaV (eingetragene Marke) Kontroller sein, der durch Fisher-Rosemount Systems vertrieben wird, und kann an verschiedene Workstations, wie beispielsweise Personalcomputer (PCs) 14 über einen Hub 16 und Ethernet-Verbindungen 18 verbunden sein. In dieser Konfiguration können die PCs 14 durch eine oder mehrere Bedienungspersonen oder Anwender verwendet werden, um mit dem Prozeßkontroller 12 eine Kommunikation herzustellen, um dadurch Informationen zu erhalten, die das Prozeßregelnetzwerk 10 betreffen, um den Status des Prozeßregelnetzwerks 10 zu überblicken oder zu ändern, um Informationen zu erhalten, die die einzelnen Bereichsvorrichtungen innerhalb des Prozeßregelnetzwerks 10 betreffen usw. Wenn der Kontroller 12 aus einem DeltaV-Kontroller besteht, kann er einen graphischen Ausschnitt der Prozeßsteuer- oder -regelroutine innerhalb des Kontrollers 12 zu dem Anwender liefern, und zwar über einen der PCs 14, wobei die Funktionsblöcke oder Steuerblöcke innerhalb der Prozeßsteueroutine veranschaulicht werden und auch die Art, in welcher diese Funktionsblöcke miteinander verkettet sind, um die Steuerung oder Regelung eines Prozesses zu bewirken. Der Anwender hat die Möglichkeit, die Prozeßsteuer- oder -regelroutine zu ändern, und zwar innerhalb des zentralisierten Prozeßreglers 12, indem er den graphischen Ausschnitt der Funktionsblöcke innerhalb der Steuer- oder Regelroutine manipuliert, um Funktionsblöcke von dieser Steueroutine wegzulassen oder zu dieser hinzuzufügen und/oder den Weg zu ändern, in welchem die Funktionsblöcke, die der Steueroutine zugeordnet sind, verkettet sind, das heißt den Weg oder die Art, in welcher sie verbunden sind.

Wie in Fig. 1 dargestellt ist, ist der zentralisierte Kontroller 12 mit zahlreichen Bereichsvorrichtungen (field devices) verbunden, die über einen Prozeß hinweg verteilt gelegen sind (allgemein durch das Bezugszeichen 19 angezeigt). Der zentralisierte Kontroller 12 kann über irgendwelche Standardtypen I/O-Karten 20 und 22 mit den Bereichsvorrichtungen 26, 28, 30, 32, 34 und 36 kommunizieren, die der zentralisierten Steuerung von dem Kontroller 12 unterworfen sind. Die I/O-Karte 20 kann beispielsweise eine analoge I/O-Karte sein, die den Kontroller 12 mit analogen Bereichsvorrichtungen 24 und 26 verbindet, die über 4-20-mA-Busse 38 kommunizieren. In ähnlicher Weise kann die I/O-Karte 22 eine digitale oder kombinierte digitale und analoge I/O-Karte sein, die mit digitalen oder gemischten digitalen und analogen Bereichsvorrichtungen in irgendeinem gewünschten Format kommuniziert, inklusive beispielsweise dem 4-20-mA-Analogformat oder irgendeinem bekannten oder standardisierten Digitalformat. Natürlich können die Bereichsvorrichtungen 26, 28, 30, 32, 34 und 36 von irgendeinem gewünschten Typ von Bereichsvorrichtungen bestehen, beispielsweise aus Sendern, Sensoren, Ventilstellgliedern, Ventilsteuervorrichtungen usw. Es sei in Verbindung mit dem Beispiel des in Fig. 1 veranschaulichten

Prozeßregelnetzwerks 10 darauf hingewiesen, daß die Bereichsvorrichtungen 26-36 Abschnitten des Prozesses 19 zugeordnet sind, die einer zentralisierten Steuerung durch die Steueroutine innerhalb des Kontrollers 12 unterworfen sind.

Der Kontroller 12 ist ebenfalls kommunikativ mit einer Interfacekarte 40 verbunden, die ihrerseits mit (oder Teil ist von) einem externen Prozeßregelnetzwerk, in welchem die Prozeßregelung in einer verteilten Weise durchgeführt wird, oder welcher Vorrichtungen enthält, die Funktionsblöcke haben, die unter Verwendung eines Kommunikationsprotokolls kommunizieren, welches verschieden ist von demjenigen, welches innerhalb des Kontrollers 12 verwendet wird. Bei der in Fig. 1 veranschaulichten Ausführungsform enthält der dezentralisierte Prozeßsteuerabschnitt des Prozesses 19 die Interfacekarte 40, einen Bus 42 und zahlreiche Bereichsvorrichtungen 43, 44, 46, 48 und 50, die an den Bus 42 angeschlossen sind. Das verteilte Prozeßregelnetzwerk von Fig. 1 kann beispielsweise ein Feldbus-Netzwerk sein, welches das Feldbus-Kommunikationsprotokoll verwendet. Wie im folgenden noch mehr in Einzelheiten erläutert werden wird, kann die Interfacekarte 40 eine aktive Verbindungsglied-Ablaufsteuerung sein, die dem Feldbus-Kommunikationsprotokoll zugeordnet ist.

Die zentralisierte Prozeßsteueroutine, die innerhalb des Kontrollers 12 gelegen ist, empfängt Eingangsgrößen von den Bereichsvorrichtungen 26-36 und möglicherweise 43-50, führt Berechnungen und andere Aktivitäten durch, die mit der Steueroutine verbunden sind, und sendet dann Befehle zu den Bereichsvorrichtungen über die I/O-Karten 20 und 22 und die Interfacekarte 40, um irgendeine gewünschte Steuerung oder Regelung des Prozesses 19 zu implementieren. Es sei jedoch darauf hingewiesen, daß der dezentralisierte Prozeßsteuer- oder -regelabschnitt des Prozeßregelnetzwerks 10 (das heißt derjenige, der dem Bus 42 in Fig. 1 zugeordnet ist) seine eigene Prozeßsteueroutine in einer dezentralisierten Weise implementieren kann, wie dies hier noch beschrieben wird, und zwar in Verbindung mit der Steuerung, die durch den Kontroller 12 ausgeführt wird. Während somit der Kontroller 12 mit einer Steuerung gekoppelt sein kann und eine gewisse Steuerung über die Vorrichtungen 43-50, die an den Bus 42 angeschlossen sind, ausführen kann, können diese Vorrichtungen auch Steuerfunktionen implementieren oder Funktionsblöcke, die mit der Steuerung verbunden sind, welche durch den Kontroller 12 implementiert wird, die jedoch statt dessen über die Vorrichtungen, die an den Bus 42 angeschlossen sind, verteilt sind.

Während bei der bevorzugten Ausführungsform der dezentralisierte Abschnitt des Prozeßregelnetzwerks 10 das Feldbus-Kommunikations- und -Steuerprotokoll verwendet, kann irgendein anderes oder gewünschtes Protokoll ebenso verwendet werden, inklusive den Protokollen, die in der Zukunft entwickelt werden. Ferner kann der Schattenfunktionsblock, der hier offenbart ist, dazu verwendet werden, um eine Kommunikation zwischen irgendwelchen zwei unterschiedlichen Steuer- oder Kommunikationsprotokollen durchzuführen, und es liegt keine Beschränkung auf die Verwendung zwischen einer zentralisierten Steueroutine und einem dezentralisierten Steuer- oder Regelnetzwerk, wie beispielsweise auf eines, welches das Feldbus-Protokoll verwendet, vor. Es kann beispielsweise zwischen zwei unterschiedlichen zentralisierten Steueroutinen oder Protokollen, die Steuerblöcke enthalten, verwendet werden. Mit anderen Worten ist der hier beschriebenen Schattenfunktionsblock nicht darauf beschränkt, mit Funktionsblöcken in dem Feldbus-Protokoll verwendet zu werden oder sogar mit einem Protokoll, welches einer dezentralisierten Steuer-

routine zugeordnet ist, sondern kann auch dazu verwendet werden, einen Controller (oder andere Vorrichtung) dazu zu befähigen, mit irgendeiner anderen externen Vorrichtung zu kommunizieren, die irgendeinen unterschiedlichen Typ eines Kommunikationsprotokolls verwendet.

Bevor Einzelheiten des Schattenfunktionsblockes der vorliegenden Erfindung erläutert werden und die Art, in welcher der zentralisierten Prozeßkontroller 12 solch einen Schattenfunktionsblock verwendet, um eine Steuerung in einem Prozeßregelnetzwerk zu implementieren, folgt eine allgemeine Beschreibung des Fieldbus-Protokolls, der Bereichsvorrichtungen, die gemäß diesem Protokoll konfiguriert sind, und die Art, in welcher die Kommunikation in dem Abschnitt des Prozeßregelnetzwerks 10 erfolgt, der das Fieldbus-Protokoll verwendet. Es sei darauf hingewiesen, daß, obwohl das Fieldbus-Protokoll bereits ein relativ neues gesamt-digitales Kommunikationsprotokoll ist, welches für die Verwendung in einem Prozeßregelnetzwerk entwickelt wurde, dieses Protokoll auf dem Gebiet bekannt ist und in Einzelheiten in vielerlei Artikeln, Broschüren und Beschreibungen beschrieben ist, die veröffentlicht, verteilt und von der Fieldbus Foundation unter anderem verfügbar sind, einer nicht auf Profit basierenden Organisation, die ihren Sitz in Austin, Texas, hat. Speziell ist das Fieldbus-Protokoll und die Art der Kommunikation mit und die Art der Speicherung von Daten in Vorrichtungen, die das Fieldbus-Protokoll verwenden, in Einzelheiten in den Bedienungsanleitungen beschrieben, mit dem Titel "Communications Technical Specification" und "User Layer Technical Specification" von der Fieldbus Foundation.

Allgemein gesagt, besteht das Fieldbus-Protokoll aus einem insgesamt digitalen, seriellen Zweiwege-Kommunikationsprotokoll, welches eine standardisierte physikalische Schnittstelle für eine Zweidrahtschleife oder Bus schafft, die "Feld"-Ausrüstungen, wie beispielsweise Sensoren, Stellglieder, Controller, Ventile usw., miteinander verbindet, die in einer Meßgeräteausrüstung oder einer Prozeßsteuer- oder -regelungsumgebung gelegen sind, beispielsweise in einer Fabrik oder Anlage. Das Fieldbus-Protokoll betrifft im Prinzip ein örtliches Bereichsnetzwerk für Bereichsmeßgeräte (Bereichsvorrichtungen) innerhalb eines Prozesses, welches diese Bereichsvorrichtungen dazu befähigt, Steuerungsfunktionen an Stellen auszuführen, die über eine Prozeßeinrichtung hinweg verteilt sind, und mit einer anderen zu kommunizieren, vor und nach der Ausführung dieser Steuerungsfunktionen, um eine Gesamtsteuer- oder -regelstrategie zu realisieren. Da das Fieldbus-Protokoll es den Steuerungsfunktionen ermöglicht, über ein Prozeßregelnetzwerk hinweg verteilt zu sein, reduziert es die Arbeitslast des zentralisierten Prozeßkontrollers für diese Bereichsvorrichtungen oder für die Bereiche des Prozesses.

Ein Prozeßsteuer- oder -regelnetzwerk, welches das Fieldbus-Protokoll verwendet, kann einen Host enthalten, der mit einer Anzahl von anderen Vorrichtungen verbunden ist, wie beispielsweise logischen Programmcontrollern, anderen Hostvorrichtungen und Bereichsvorrichtungen (field devices) über eine Zweidraht-Fieldbus-Schleife oder -Bus, wie beispielsweise den Bus 42 von Fig. 1. Der Fieldbus-Bus kann unterschiedliche Abschnitte oder Segmente enthalten, die durch Brückenvorrichtungen getrennt sind, wobei jeder Abschnitt des Busses einen untergeordneten Satz der Vorrichtungen verbindet, die an den Bus angebracht sind, um Kommunikationen zwischen den Vorrichtungen in einer Weise zu ermöglichen, wie sie im folgenden beschrieben wird. In typischer Weise ist eine Konfiguriervorrichtung in einer der Vorrichtungen gelegen, wie beispielsweise einem Host, und ist für die Einrichtung oder Konfigurierung jeder der Vorrichtungen verantwortlich (die "Smart"-Vorrichtun-

gen insofern sind, als sie jeweils einen Mikroprozessor enthalten, der eine Kommunikation durchführen kann und in einigen Fällen Steuerungsfunktionen durchführen kann), der aber ebenso erkennen kann, wenn neue Bereichsvorrichtungen an den Fieldbus-Bus angeschlossen werden, wenn die Bereichsvorrichtungen von dem Bus entfernt werden, der Daten erkennen kann, die durch die Bereichsvorrichtungen, welche an den Bus angeschlossen sind, erzeugt werden, und der eine Kopplung mit einer oder mit mehreren Anwenderterminals bewirkt, die in dem Host oder irgendeiner anderen Vorrichtung gelegen sein können, welche in irgendeiner Weise an den Host angeschlossen sind.

Der Fieldbus-Bus unterstützt oder erlaubt zweierlei: eine rein digitale Kommunikation und er kann auch ein Stromversorgungssignal zu irgendeiner oder zu all den Vorrichtungen, die an ihn angeschlossen sind, übertragen, wie beispielsweise an die Bereichsvorrichtungen. Alternativ kann irgendeine oder können alle Fieldbus-Vorrichtungen ihre eigene Stromversorgung besitzen oder können über separate Leitungen mit externen Stromversorgungen verbunden sein. Das Fieldbus-Protokoll erlaubt es vielen Vorrichtungen/Verdrahtungstopologien, inklusive Vielfachvorrichtungen, an das gleiche Paar von Drähten oder Leitungen angeschlossen zu werden, erlaubt Punkt-zu-Punkt-Verbindungen, in welchen jede Vorrichtung an einen Controller oder einen Host über ein getrenntes Zweidrahtpaar angeschlossen ist (ähnlich den typischen 4-20 mA Analogsystemen) und Baum- oder "spur"-Verbindungen realisiert sind, in denen jede Vorrichtung mit einem gemeinsamen Punkt in einem Zweileitungsbus verbunden ist, der beispielsweise aus einem Kabelkasten oder Verzweigungsstück oder einem Abschlußbereich in einer der Bereichsvorrichtungen innerhalb eines Prozeßsteuer- oder -regelnetzwerks bestehen kann.

Es können Daten über die Bussegmente in irgendeiner einer Anzahl von unterschiedlichen Kommunikationsbauraten oder Geschwindigkeiten gemäß dem Fieldbus-Protokoll gesendet werden. Beispielsweise schafft das Fieldbus-Protokoll eine 31,25-Kbit/s-Kommunikationsrate (H1) oder eine 1,0-Mbit/s- und/oder eine 2,5-Mbit/s-(H2)-Kommunikationsrate, die in typischer Weise für eine fortschrittliche Prozeßsteuerung oder -regelung, Ferneingabe-/ausgabe und für Hochgeschwindigkeits-Fabrikautomatisationsanwendungen angewendet werden. In gleicher Weise können über den Fieldbus-Bus Daten unter Verwendung einer Spannungsmod-Signalgebung oder einer Strommod-Signalgebung übertragen werden. Die maximale Länge von irgendeinem Fieldbus-Bus oder -Segment ist nicht strikt begrenzt, sondern wird statt dessen durch die Kommunikationsrate, den Kabeltyp, die Leitungsgröße, Busleistungsoptionen usw. festgelegt.

Das Fieldbus-Protokoll klassifiziert die Vorrichtungen, die an den Bus angeschlossen werden können, in drei primäre Kategorien, nämlich die Basisvorrichtungen, die Verbindungsmastervorrichtungen und die Brückenvorrichtungen. Die Basisvorrichtungen können kommunizieren, das heißt Kommunikationssignale auf den Bus senden oder von diesem empfangen, sind jedoch nicht dazu befähigt, die Reihenfolge oder Zeitsteuerung der Kommunikation zu steuern, die auf dem Bus auftreten. Die Verbindungsmastervorrichtungen bestehen aus Vorrichtungen, die über dem Bus kommunizieren und die Fähigkeit haben, den Fluß und die Zeitsteuerung der Kommunikationssignale auf den Bus zu steuern. Die Brückenvorrichtungen bestehen aus Vorrichtungen, die dafür konfiguriert sind, um mit einzelnen Segmenten oder Zweigen eines Fieldbus-Busses zu kommunizieren oder diese einzelnen Segmente oder Zweige miteinander zu verbinden, um ein größere Prozeßsteuer- oder -regelnetzwerk zu erzeugen. Wenn es gewünscht wird, können die

Brückenvorrichtungen zwischen unterschiedlichen Datengeschwindigkeiten und/oder unterschiedlichen Datensignalisierungsformaten eine Umwandlung vornehmen, die auf den unterschiedlichen Segmenten des Busses verwendet werden, können Signale verstärken, die zwischen den Segmenten des Busses wandern, können die Signale filtern, die zwischen unterschiedlichen Segmenten des Busses verlaufen, und können lediglich solche Signale durchlassen, die dafür bestimmt sind, durch eine Vorrichtung an einem der Bussegmente empfangen zu werden, an welches die Brücke gekoppelt ist und/oder können andere Aktionen vornehmen, die erforderlich sind, um unterschiedliche Segmente des Busses zu verbinden. Die Brückenvorrichtungen, welche die Bussegmente verbinden, die mit unterschiedlichen Geschwindigkeiten arbeiten, müssen Verbindungsmasterfähigkeiten auf der Seite des Segmentes mit niedrigerer Geschwindigkeit der Brücke besitzen.

Jede der Fieldbus-Vorrichtungen besitzt die Fähigkeit, über den Bus zu kommunizieren, und, was wichtig ist, die Fähigkeit unabhängig eine oder mehrere Prozeßsteuerfunktionen durchzuführen unter Verwendung von Daten, die durch die Vorrichtung von dem Prozeß erworben wurden, oder von einer unterschiedlichen Vorrichtung erworben wurden, und zwar über die Kommunikationssignale auf dem Bus. Die Fieldbus-Vorrichtungen besitzen daher die Fähigkeit, direkt Abschnitte einer Gesamtsteuer- oder -regelstrategie zu implementieren, die in der Vergangenheit durch einen zentralisierten digitalen Controller durchgeführt wurden. Um Steuerfunktionen durchzuführen, enthält jede Fieldbus-Vorrichtung einen oder mehrere standardisierte "Blöcke", die in einem Mikroprozessor innerhalb der Vorrichtung implementiert sind. Speziell enthält jede Fieldbus-Vorrichtung einen Ressourcenblock, Null- oder Mehrfunktionsblöcke und Null- oder Mehrwandlerblöcke. Diese Blöcke werden als Blockobjekte bezeichnet.

Ein Ressourcenblock speichert und überträgt spezifische Vorrichtungsdaten, die einige Charakteristika einer Fieldbus-Vorrichtung betreffen, inklusive beispielsweise einem Vorrichtungstyp, einer Vorrichtungsrevisionsanzeige und Anzeigen darüber, wo andere spezifische Vorrichtungsinformationen innerhalb eines Speichers der Vorrichtung erhalten werden können. Während verschiedene Vorrichtungshersteller unterschiedliche Typen der Daten in dem Ressourcenblock einer Bereichsvorrichtung speichern können, enthält jede Bereichsvorrichtung, die mit dem Fieldbus-Protokoll konform ist, einen Ressourcenblock, der einige Daten speichert.

Ein Funktionsblock definiert und implementiert eine Eingabefunktion, eine Ausgabefunktion oder eine Steuerfunktion, die der Bereichsvorrichtung zugeordnet ist und es werden somit Funktionsblöcke allgemein als Eingabe-, Ausgabe- und Steuerfunktionsblöcke bezeichnet. Es können jedoch andere Kategorien von Funktionsblöcken existieren, wie beispielsweise Hybrid-Funktionsblöcke, oder können auch in der Zukunft entwickelt werden. Jeder Eingabe- oder Ausgabefunktionsblock erzeugt wenigstens eine Prozeßsteuereingangsgroße (wie beispielsweise eine Prozeßvariable aus einer Prozeßmeßvorrichtung) oder eine Prozeßsteuerausgangsgroße (wie beispielsweise eine Ventilposition, die zu einer Stellvorrichtung gesendet wird), während jeder Steuerfunktionsblock einen Algorithmus verwendet (der in seiner Natur geschützt oder firmeneigen sein kann), um eine oder mehrere Prozeßausgangsgroßen aus einer oder aus mehreren Prozeßeingangsgroßen und aus Steuereingangsgroßen zu erzeugen. Beispiele von Standardfunktionsblöcken umfassen Analogeingangs(AI)-, Analogausgangs(AO)-, Vorspann(B)-, Steuerwähl(CS)-, Diskreteingangs(DI)-, Diskreteausgangs(DO)-, Handlade(ML)-, Proportional/Dif-

ferential(PD)-, Proportional/Integral/Differential(PDI)-, Verhältnis(RA)- und Signalwähl(SS)-Funktionsblöcke. Es existieren jedoch andere Typen von Funktionsblöcken und neue Typen von Funktionsblöcken können festgelegt oder hergestellt werden, um in der Fieldbus-Umgebung zu arbeiten.

Ein Wandlerblock koppelt die Eingangsgrößen und Ausgangsgrößen eines Funktionsblockes an die örtlichen Hardwarevorrichtungen, wie beispielsweise an Sensoren und Vorrichtungsteilglieder, um die Funktionsblöcke dazu zu befähigen, die Ausgangsgrößen von örtlichen Sensoren zu lesen und die örtlichen Vorrichtungen mit Befehlen zu versehen, um eine oder mehrere Funktionen, wie beispielsweise das Bewegen eines Ventils auszuführen. Wandlerblöcke enthalten in typischer Weise Informationen, die erforderlich sind, um Signale zu interpretieren, die durch eine örtliche Vorrichtung abgeleitet wurden, und um in richtiger Weise die örtlichen Hardwarevorrichtungen zu steuern, wobei diese Informationen beispielsweise Information enthalten, die den Typ einer örtlichen Vorrichtung identifizieren, Eichungsinformationen, die einer örtlichen Vorrichtung zugeordnet sind, betreffen usw. Ein einzelner Wandlerblock ist in typischer Weise jedem Eingabe- oder Ausgabefunktionsblock zugeordnet.

Die meisten Funktionsblöcke sind dazu befähigt, Alarm oder Ereignisanzeigen zu generieren, basierend auf vorbestimmten Kriterien, und haben die Fähigkeit unterschiedlich in verschiedenen Modi zu arbeiten. Allgemein gesagt, können Funktionsblöcke in einem automatischen Modus arbeiten, in welchem beispielsweise der Algorithmus eines Funktionsblockes automatisch arbeitet; in einem Operatormodus arbeiten, in welchem der Eingang oder der Ausgang eines Funktionsblockes von Hand gesteuert wird; einem Außer-Service-Modus arbeiten, in welchem der Block nicht arbeitet; in einem Kaskadenmodus arbeiten, in welchem die Operation des Blockes von der Ausgangsgröße eines verschiedenen Blockes beeinflusst wird (durch diesen bestimmt wird); und in einem oder mehreren Entfernungsmodi arbeiten, in welchen ein entfernt gelegener Computer den Modus des Blocks bestimmt. Es existieren jedoch in dem Fieldbus-Protokoll andere Betriebsmodi.

Als wichtiges Merkmal besitzt jeder Block die Fähigkeit, mit anderen Blöcken in der gleichen oder unterschiedlichen Bereichsvorrichtungen über den Fieldbus-Bus zu kommunizieren, und zwar unter Verwendung von Standardnachrichtenformaten, die durch das Fieldbus-Protokoll festgelegt sind. Als ein Ergebnis können Kombinationen von Funktionsblöcken (in den gleichen oder unterschiedlichen Vorrichtungen) miteinander kommunizieren, um eine oder mehrere dezentralisierte Regelschleifen zu erzeugen. Es kann somit beispielsweise ein PID-Funktionsblock in einer Bereichsvorrichtung über den Fieldbus-Bus angeschlossen sein, um eine Ausgangsgröße eines AI-Funktionsblockes in einer zweiten Bereichsvorrichtung zu empfangen, um Daten an einen AO-Funktionsblock in einer dritten Bereichsvorrichtung zu liefern, um eine Ausgangsgröße eines AO-Funktionsblockes als Rückkopplungsgröße zu empfangen, um eine Prozeßregelschleife getrennt und neben irgendeinem zentralisierten Controller zu erzeugen. Auf diese Weise bewegen Kombinationen von Funktionsblöcken die Steuerfunktionen aus einer zentralisierten DCS-Umgebung heraus, was die Möglichkeit schafft, daß DCS-Vielfunktionskontrollen Überwachungs- oder Koordinationsfunktionen durchführen. Ferner erlauben es die Funktionsblöcke, eine graphische blockorientierte Struktur zu verwenden, und zwar eine einfache Konfiguration eines Prozesses, und ermöglichen die Verteilung der Funktionen unter den Bereichsvorrichtungen von unterschiedlichen Zulieferern, da

diese Blöcke ein konsistentes Kommunikationsprotokoll verwenden.

Zusätzlich zu dem Merkmal, daß Bereichsvorrichtungen Blockobjekte enthalten und implementieren, enthält jede Bereichsvorrichtung ein oder mehrere andere Objekte, inklusive Verbindungsobjekten, Trendobjekten, Warnobjekten und Betrachtungsobjekten. Verbindungsobjekte definieren die Verbindungen zwischen den Eingängen und Ausgängen der Blöcke (wie den Funktionsblöcken), und zwar sowohl intern in der Bereichsvorrichtung als auch über den Fieldbus-Bus hinweg. Trendobjekte erlauben einen örtlichen Trend von Funktionsblockparametern für den Zugriff auf andere Vorrichtungen, wie beispielsweise einen Host oder einen Kontrolller. Trendobjekte halten historische Kurzzeitdaten, die einige, beispielsweise den Funktionsblockparameter, betreffen und berichten diese Daten an andere Vorrichtungen oder Funktionsblöcke, und zwar über den Fieldbus-Bus in einer asynchronen Weise. Warnobjekte berichten über Alarmer und Ereignisse über den Fieldbus-Bus. Diese Alarmer oder Ereignisse können irgendein Ereignis betreffen, welches innerhalb einer Vorrichtung stattfindet oder innerhalb einem der Blöcke einer Vorrichtung. Betrachtungsobjekte sind vordefinierte Gruppierungen von Blockparametern, die in standardisierten Mensch/Maschine-Schnittstellenbereichen verwendet werden und die zu anderen Vorrichtungen gesendet werden können, und zwar unter Verwendung von asynchronen Übertragungen, um eine Betrachtung von Zeit zu Zeit zu ermöglichen.

In Fig. 2 sind drei Fieldbus-Vorrichtungen veranschaulicht, die beispielsweise aus irgendeiner der Bereichsvorrichtungen 43-50 von Fig. 1 bestehen, und diese enthalten Ressourcenblöcke 58, Funktionsblöcke 60, 61 oder 62 und Wandlerblöcke 63 und 64. In der ersten Vorrichtung ist der Funktionsblock 60 (der aus einem Eingabefunktionsblock besteht) über den Wandlerblock 63 mit einem Sensor 65 gekoppelt, der beispielsweise aus einem Temperatursensor, einem Sollpunktanzeigsensor usw. bestehen kann. In der zweiten Vorrichtung ist der Funktionsblock 61 (der aus einem Ausgabefunktionsblock bestehen kann) über den Wandlerblock 64 an eine Ausgabevorrichtung, wie beispielsweise ein Ventil 66, gekoppelt. In der dritten Vorrichtung enthält der Funktionsblock 62 (der aus einem Steuerfunktionsblock bestehen kann) ein Trendobjekt 67, welches diesem zugeordnet ist, um den Trend des Eingangsparameters des Funktionsblocks 62 zu ermitteln.

Die Verbindungsobjekte 68 definieren die Verbindungen von Blockparametern von jedem der zugeordneten Blöcke und die Warnobjekte 69 erzeugen Alarmer oder Ereignisbenachrichtigungen für jeden der zugeordneten Blöcke. Betrachtungsobjekte 70 sind mit jedem der Funktionsblöcke 60, 61 und 62 zugeordnet und enthalten Gruppendatenlisten für die Funktionsblöcke, zu denen sie zugeordnet sind. Diese Listen enthalten Informationen, die für jeden eines Satzes von unterschiedlich definierten Betrachtungen erforderlich sind. Natürlich stellen die Vorrichtungen von Fig. 2 lediglich Beispiele dar und andere Zahlen und Typen von Blockobjekten, Verbindungsobjekten, Warnobjekten, Trendobjekten und Betrachtungsobjekten können in irgendeiner Bereichsvorrichtung vorgesehen sein.

Um eine Kommunikation und Steueraktivitäten zu implementieren und auszuführen, verwendet das Fieldbus-Protokoll drei allgemeine Kategorien der Technologie, die als eine physikalische Schicht, als ein Kommunikations-"Stapel" und eine Anwenderschicht definiert sind. Die Anwenderschicht enthält die Steuer- und Konfigurationsfunktionen, die in Form der Blöcke vorgesehen werden (wie beispielsweise den Funktionsblöcken) und Objekte innerhalb irgendeiner bestimmten Prozeßsteuervorrichtung oder Be-

reichsvorrichtung. Die Anwenderschicht ist in typischer Weise in einer geschützten Weise ausgelegt oder konstruiert, und zwar durch den Hersteller der Vorrichtung, muß jedoch die Fähigkeit haben, Nachrichten zu empfangen und auszusenden, und zwar in Einklang mit dem Standardnachrichtenformat, welches durch das Fieldbus-Protokoll festgelegt ist, und muß durch einen Anwender in der standardisierten Weise konfigurierbar sein. Die physikalische Schicht und der Kommunikationsstapel sind erforderlich, um eine Nachrichtenverbindung zwischen unterschiedlichen Blöcken unterschiedlicher Feld- oder Bereichsvorrichtungen in einer standardisierten Weise zu bewirken, und zwar unter Verwendung eines Zweidrahtbusses, und können durch das gut bekannte Open-Systems-Interconnect-(OSI)-Schicht-Kommunikationsmodell modelliert sein.

Die physikalische Schicht, die der OSI-Schicht 1 entspricht, ist in jeder Bereichsvorrichtung und in dem Bus eingebettet und arbeitet dahingehend, um elektromagnetische Signale, die von dem Fieldbus-Übertragungsmedium (dem Fieldbus-Zweidrahtbus) empfangen werden, in Nachrichten umzusetzen, die durch den Kommunikationsstapel der Bereichsvorrichtung verwendet werden können. Die physikalische Schicht kann als ein Fieldbus-Bus und elektromagnetischen Signalen, die auf dem Bus an den Eingängen und Ausgängen der Bereichsvorrichtungen vorhanden sind, betrachtet werden.

Der Kommunikationsstapel, der in jeder Fieldbus-Vorrichtung vorhanden ist, enthält eine Datenverbindungsschicht, die der OSI-Schicht 2 entspricht, eine Fieldbus-Zugriffs-Sub-Schicht und eine Fieldbus-Nachricht-Spezifikationsschicht. Die Anwenderschicht des Fieldbus-Protokolls besteht aus einer Schicht, die in dem OSI-Protokoll nicht definiert ist. Jede Schicht in dem Kommunikationsstapel ist für die Kodierung und Dekodierung eines Abschnitts der Nachricht oder des Signals verantwortlich, welches über den Fieldbus-Bus übertragen wird. Als ein Ergebnis addiert oder entfernt jede Schicht des Kommunikationsstapels gewisse Abschnitte des Fieldbus-Signals, wie beispielsweise die Präambeln, Startbegrenzer und Endebegrenzer, und in einigen Fällen, dekodiert sie Bandabschnitte des Fieldbus-Signals, um eine Identifizierung darüber durchzuführen, ob der Rest des Signals oder der Nachricht gesendet werden sollte oder ob das Signal gelöscht werden sollte, da es beispielsweise eine Nachricht oder Daten für Funktionsblöcke enthält, die nicht innerhalb der empfangenden Bereichsvorrichtung vorhanden sind.

Die Datenverbindungsschicht steuert die Übertragung von Nachrichten auf dem Fieldbus-Bus und managt den Zugriff auf den Bus gemäß einer deterministischen zentralisierten Busablaufsteuerung (scheduler), der als ein verbindungsaktiver Abwickler bezeichnet wird, der im folgenden noch mehr in Einzelheiten beschrieben wird. Die Datenverbindungsschicht entfernt eine Präambel von den Signalen auf dem Übertragungsmedium und kann die empfangene Präambel verwenden, um den internen Takt der Bereichsvorrichtung mit dem ankommenden Fieldbus-Signal zu synchronisieren. In ähnlicher Weise wandelt die Datenverbindungsschicht die Nachrichten auf dem Kommunikationsstapel in physikalische Fieldbus-Signale um und kodiert diese Signale mit der Taktinformation, um ein "synchrones serielles" Signal mit einer richtigen Präambel für die Übertragung auf dem Zweidrahtbus oder -schleife zu erzeugen. Während des Dekodierungsprozesses erkennt die Datenverbindungsschicht spezifische Codes innerhalb der Präambel, wie beispielsweise Startbegrenzer und Endebegrenzer, um den Anfang und das Ende einer bestimmten Fieldbus-Nachricht zu identifizieren, und kann eine Prüfsumme erstellen, um die Integrität des Signals oder der Nachricht, die von dem Bus

empfangen wurde, zu verifizieren. In ähnlicher Weise überträgt die Datenverbindungsschicht die Fieldbus-Signale auf den Bus, indem sie Start- und Endegrenzen zu den Nachrichten auf dem Datenkommunikationsstapel hinzufügt und indem sie diese Signale auf das Übertragungsmedium zu einer geeigneten Zeit setzt:

Die Fieldbus-Nachrichten-Spezifikationsschicht erlaubt es der Anwenderschicht (das heißt den Funktionsblöcken, Objekten usw. einer Bereichsvorrichtung), über den Bus zu kommunizieren, und zwar unter Verwendung eines Standardsatzes von Nachrichtenformaten und beschreibt den Kommunikationsservice, die Nachrichtenformate und die Protokollverhaltensweisen, die erforderlich sind, um Nachrichten herzustellen, die auf den Kommunikationsstapel zu setzen sind und die für die Anwenderschicht vorzusehen sind. Da die Fieldbus-Nachrichten-Spezifikationsschicht standardisierte Kommunikationen für die Anwenderschicht liefert, sind spezifische Fieldbus-Nachrichten-Spezifikations-Kommunikationsdienste für jeden Typ des oben beschriebenen Objekts festgelegt. Beispielsweise enthält die Fieldbus-Nachrichten-Spezifikationsschicht Objektwörterbuchdienste, die es dem Anwender ermöglichen, in einem Objektwörterbuch einer Vorrichtung zu lesen. Das Objektwörterbuch speichert Objektbeschreibungen, welche jedes der Objekte beschreiben oder identifizieren (wie beispielsweise Blockobjekte), und zwar von einer Vorrichtung. Die Fieldbus-Nachrichten-Spezifikationsschicht ermöglicht auch variable Zugriffsdienste, die es einem Anwender ermöglichen, Kommunikationsbeziehungen, die als virtuelle Kommunikationsbeziehungen (VCRs) bekannt sind und im folgenden beschrieben werden, die einer oder mehreren Objekten einer Vorrichtung zugeordnet sind, zu lesen und zu ändern. Darüber hinaus schafft die Fieldbus-Nachrichten-Spezifikationsschicht ein Kontextmanagement, variable Zugriffsdienste, Ereignisdienste, Hochlade- und Herunterladedienste und Programmaufrufdienste, die alle in Verbindung mit dem Fieldbus-Protokoll gut bekannt sind und daher hier nicht in Einzelheiten beschrieben werden. Die Fieldbus-Zugriffs-Sub-Schicht bildet die Fieldbus-Nachrichten-Spezifikationsschicht in der Datenverbindungsschicht ab.

Um einen Betrieb dieser Schichten zuzulassen oder zu ermöglichen, enthält jede Fieldbus-Vorrichtung eine Managementinformationsbasis (MIB), die aus einer Datenbasis oder Datenbank besteht, die VCRs, dynamische Variable, statistische Größen, Zeitsteuerpläne, Funktionsblockausführungszeitsteuerpläne und eine Vorrichtungsetikette (device tag) und Adresseninformationen speichert. Natürlich können die Informationen innerhalb der MIB zugegriffen werden oder zu irgendeiner Zeit unter Verwendung der Standard-Fieldbus-Nachrichten oder -Befehle geändert werden. Ferner ist gewöhnlich eine Vorrichtungsbeschreibung für jede Vorrichtung vorgesehen, um einem Anwender oder einem Host einen weiten Überblick an Informationen in der VFD zu geben. Eine Vorrichtungsbeschreibung, die in typischer Weise in Form einer Sendeerlaubnis, um von einem Host verwendet zu werden, festgelegt ist, speichert Informationen, die für den Host erforderlich sind, um die Bedeutung der Daten in den VFDs einer Vorrichtung zu verstehen.

Wie ersehen werden kann, muß die Implementierung von irgendeiner Steuerstrategie unter Verwendung der Funktionsblöcke, die über ein Prozeßregelnetzwerk verteilt angeordnet sind, die Ausführung der Funktionsblöcke präzise geplant werden, und zwar in bezug auf die Ausführung von anderen Funktionsblöcken in einer bestimmten Regelschleife. In ähnlicher Weise muß die Kommunikation zwischen unterschiedlichen Funktionsblöcken in präziser Weise auf dem Bus geplant werden, so daß die richtigen Daten zu jedem Funktionsblock übertragen werden, bevor dieser

Block ein Programm ausführt.

Der Weg, mit welchem unterschiedliche Bereichsvorrichtungen (und unterschiedliche Blöcke innerhalb den Bereichsvorrichtungen) über das Fieldbus-Übertragungsmedium kommunizieren, soll nun beschrieben werden. Damit eine Kommunikation stattfindet, arbeitet eine der Verbindungsmastervorrichtungen auf jedem Segment der Fieldbus-Schleife als ein verbindungsaktiver Abwickler (LAS), der aktiv die Kommunikation auf dem zugeordneten Segment des Busses abwickelt und steuert. Das LAS für jedes Segment des Busses speichert einen Kommunikationsplan (einen verbindungsaktiven Plan), der Zeitpunkte enthält, bei denen jeder Funktionsblock von jeder Vorrichtung geplant ist, periodisch die Kommunikationsaktivität auf dem Bus zu starten und auch die Länge der Zeit geplant ist, während welcher diese Kommunikationsaktivität stattfinden muß. Während lediglich eine und nur eine aktive LAS-Vorrichtung auf jedem Segment des Busses vorhanden sein kann, können andere Verbindungsmastervorrichtungen als Backup-LASs dienen und können beispielsweise dann aktiv werden, wenn die momentane LAS ausfällt. Die Basisvorrichtungen haben nicht die Fähigkeit, eine LAS zu irgendeinem Zeitpunkt zu werden.

Allgemein gesagt, werden die Kommunikationsaktivitäten über den Bus eingeteilt in Wiederholungsmakrozyklen, die den synchronen Kommunikationsplan für den Bus definieren. Eine Vorrichtung kann aktiv sein, das heißt Daten zu irgendeinem Segment des Busses senden oder von diesem empfangen, und zwar selbst wenn sie physikalisch an ein unterschiedliches Segment des Busses angeschlossen ist, und zwar durch eine koordinierte Operation der Brücken und der LASs auf dem Bus.

Während jedes Makrozyklus führt jeder der Funktionsblöcke, die auf einem bestimmten Segment des Busses aktiv sind, gewöhnlich zu einem unterschiedlichen, jedoch präzise geplanten (synchrone) Zeitpunkt oder Zeit ein Programm durch. Wenn der Funktionsblock einen Ausgangsparameter hat, der mit einem anderen Parameter extern zu der Vorrichtung verkettet ist, so veröffentlicht der Funktionsblock seine Ausgangsdaten in einer präzise geplanten Zeit im Ansprechen auf einen Erzwungsdatenbefehl, der durch die LAS erzeugt wurde. In bevorzugter Weise ist jeder Funktionsblock so eingeplant, daß er seine Ausgabedaten kurz nach dem Ende der Ausführungsperiode des Funktionsblockes veröffentlicht. Ferner sind die Datenveröffentlichungszeitpunkte der verschiedenen Funktionsblöcke in serieller Form geplant oder einer Ablaufsteuerung unterworfen, so daß keine zwei Funktionsblöcke auf einem bestimmten Segment des Busses Daten zur gleichen Zeit veröffentlichen. Während der Zeit, während welcher eine synchrone Kommunikation nicht stattfindet, hat jede Bereichsvorrichtung die Möglichkeit, ihrerseits Alarmdaten, Betrachtungsdaten usw. in einer asynchronen Weise zu senden, und zwar unter Verwendung von sendeanfrage-angetriebenen Kommunikationen. Der Ausführungs- oder Ablaufplan ist in einer Managementinformationsbasis (MIB) abgespeichert, es sind die Ausführungszeitpunkte der Blöcke in den Funktionsblöcken gespeichert und es sind die Zeitpunkte zum Senden der Zwangsdatenbefehle zu jeder der Vorrichtungen auf einem Segment des Busses in der MIB der LAS-Vorrichtung für dieses Segment gespeichert. Diese Zeitpunkte werden in typischer Weise als Offsetzeiten gespeichert, da sie die Zeiten oder Zeitpunkte identifizieren, zu welchen Funktionsblock ein Programm ausführen oder Daten senden muß, und zwar in einer Versetzung (Offset) vom Anfang einer "absoluten Verbindungsplanstartzeit", die allen Vorrichtungen, die an den Bus angeschlossen sind, bekannt ist.

Um Kommunikationen zwischen jedem Makrozyklus zu

bewirken, sendet die LAS einen Zwangsdatenbefehl zu jeder der Vorrichtungen auf dem zugeordneten Bussegment gemäß der Liste der Sendezeitpunkte, die in dem verbindungsaktiven Plan gespeichert sind. Nach dem Empfang eines Zwangsdatenbefehls veröffentlicht ein Funktionsblock einer Vorrichtung seine Ausgabedaten auf dem Bus. Da jeder der Funktionsblöcke typischerweise einer Ablaufsteuerung unterliegt, um ein Programm auszuführen, so daß die Ausführung des Programms dieses Blocks vervollständigt wird, kurz bevor der Block im Plan dafür vorgesehen ist, einen Zwangsdatenbefehl zu empfangen, sollten die Daten, die im Ansprechen auf einen Zwangsdatenbefehl veröffentlicht werden, die allerletzten Ausgangsdaten des Funktionsblockes sein. Wenn jedoch ein Funktionsblock in der Programmausführung langsam ist und keine neuen Ausgangsgrößen verriegelt hat, wenn er den Zwangsdatenbefehl empfängt, veröffentlicht der Funktionsblock die Ausgangsdaten, die während des letzten Laufes des Funktionsblockes erzeugt wurden.

Während der Perioden, in welchen die Zwangsdatenveröffentlichungen nicht geplant sind, kann die LAS asynchrone Kommunikationsaktivitäten aktivieren. Um eine asynchrone Kommunikation zu bewirken, sendet die LAS eine Durchlaß-Token-Nachricht zu einer bestimmten Bereichsvorrichtung. Wenn eine Bereichsvorrichtung eine Durchlaß-Token-Nachricht empfängt, erlangt diese Bereichsvorrichtung vollen Zugriff auf den Bus (oder ein Segment desselben) und kann asynchron Nachrichten senden, wie beispielsweise Alarmanmeldungen, Trenddaten, Operator-Sollpunktänderungen usw., bis die Nachrichten vervollständigt sind oder bis eine maximal zugewiesene "Token-Haltezeit" verstrichen ist. Danach gibt die Feldvorrichtung den Bus frei (oder irgendein bestimmtes Segment desselben) und die LAS sendet eine Durchlaß-Token-Nachricht zu einer anderen Vorrichtung. Dieser Prozeß wird wiederholt, bis die LAS entweder gemäß dem Plan einen Zwangsdatenbefehl aussendet, um eine synchrone Kommunikation zu bewirken, oder die LAS eine Netzwerkwartung durchzuführen hat. Natürlich kann abhängig vom Ausmaß des Nachrichtenverkehrs und der Zahl der Vorrichtung und der Blöcke, die an ein bestimmtes Segment des Busses gekoppelt sind, nicht jede Vorrichtung eine Durchlaß-Token-Nachricht während jedes Makrozyklusses empfangen.

Fig. 3 veranschaulicht ein Zeitsteuerungschema, welches die Zeitpunkte herausgreift, bei den die Funktionsblöcke (mit AI_{LOOP1} , PID_{LOOP1} , AI_{LOOP2} , AO_{LOOP1} , SS_{LOOP2} und PID_{LOOP3}) ein Programm ausführen, und zwar während jedes Makrozyklusses eines Bussegmentes, und die Zeitpunkte, bei denen eine synchrone Kommunikation während jedes Makrozyklusses stattfindet, der dem Bussegment zugeordnet ist. In dem Zeitsteuerplan von Fig. 3 ist die Zeit auf der horizontalen Achse aufgetragen und die Aktivitäten, die den unterschiedlichen Funktionsblöcken zugeordnet sind, sind auf der vertikalen Achse veranschaulicht. Die Regelschleife (die zum Zwecke der Erläuterung willkürlich ist), in welcher jeder der Funktionsblöcke arbeitet, ist in Fig. 3 als Indexangabe identifiziert. Somit verweist AI_{LOOP1} auf den AI -Funktionsblock von beispielsweise einem Sender, der innerhalb einer ersten Regelschleife arbeitet, PID_{LOOP1} verweist auf den PID -Funktionsblock in beispielsweise einem Stellwerk/Ventil, welches innerhalb der ersten Regelschleife arbeitet, usw. Die Blockausführungsperiode von jedem der veranschaulichten Funktionsblöcke ist durch ein kreuzstrichliertes Kästchen herausgestellt, während jede geplante synchrone Kommunikation durch einen vertikalen Balken in Fig. 3 identifiziert ist.

Somit führt gemäß dem Zeitsteuerplan von Fig. 3 während irgendeines bestimmten Makrozyklusses des Busseg-

ments der AI_{LOOP1} -Funktionsblock zuerst ein Programm aus, und zwar für die Zeitperiode, die durch das Kästchen 71 spezifiziert ist. Dann, während der Zeitperiode, die durch den vertikalen Balken 72 angezeigt ist, wird die Ausgangsgröße des AI_{LOOP1} -Funktionsblockes auf dem Bussegment im Ansprechen auf einen Zwangsdatenbefehl von der LAS für das Bussegment veröffentlicht. In ähnlicher Weise zeigen die Kästchen 74, 76, 78, 80 und 80 die Ausführungszeiten der jeweiligen Funktionsblöcke PID_{LOOP1} , AI_{LOOP2} , AO_{LOOP1} , SS_{LOOP2} und PID_{LOOP3} , die für jeden der unterschiedlichen Blöcke verschieden sind, an, während die vertikalen Balken 82, 84, 86, 88 und 89 die Zeiten anzeigen, zu welchen die jeweiligen Funktionsblöcke PID_{LOOP1} , AI_{LOOP2} , AO_{LOOP1} , SS_{LOOP2} und PID_{LOOP3} Daten auf dem Bussegment veröffentlichen.

Wie zu erkennen ist, veranschaulicht das Zeitsteuerungschema von Fig. 3 auch die Zeitpunkte oder Zeiten, die für asynchrone Kommunikationsaktivitäten verfügbar sind, die während der Ausführungszeiten von irgendeinem der Funktionsblöcke auftreten können und während der Zeit am Ende des Makrozyklusses, während welcher keiner der Funktionsblöcke ein Programm ausführt und wenn keine synchrone Kommunikation auf dem Bussegment stattfindet. Wenn es natürlich gewünscht wird, können unterschiedliche Funktionsblöcke beabsichtigt so eingeplant werden, daß sie ein Programm zur gleichen Zeit ausführen und daß nicht alle Funktionsblöcke Daten auf dem Bus veröffentlichen müssen, wenn beispielsweise keine andere Vorrichtung an den Datenteil hat, die durch einen Funktionsblock erzeugt werden.

Bereichsvorrichtungen können Daten veröffentlichen oder übertragen und können Nachrichten über den Feldbus-Bus übertragen, und zwar unter Verwendung von einer der drei virtuellen Kommunikationsbeziehungen (VCRs), die in der Feldbus-Zugriffs-Sub-Schicht des Stapels von jeder Bereichsvorrichtung definiert sind. Ein Client/Server-VCR wird für in eine Warteschlange eingereihte, nicht geplante, vom Anwender initialisierte Eins-zu-Eins-Kommunikationen zwischen den Vorrichtungen auf dem Bus verwendet. Solche in eine Warteschlange eingereichten Nachrichten werden in der Reihenfolge gesendet und empfangen, wie sie für die Aussendung geliefert werden, und zwar gemäß deren Priorität, ohne daß dabei frühere Nachrichten überschrieben werden. Somit kann eine Bereichsvorrichtung einen Client/Server-VCR verwenden, wenn sie eine Durchlaß-Token-Nachricht von einem LAS empfängt, um eine Anfragenachricht zu einer anderen Vorrichtung auf dem Feldbus-Bus zu senden. Der Anfrager wird als "Client" bezeichnet und die Vorrichtung, die die Anfrage empfängt, wird als "Server" bezeichnet. Der Server sendet eine Antwort, wenn er eine Durchgangs-Token-Nachricht von dem LAS empfängt. Der Client/Server-VCR wird beispielsweise dazu verwendet, um operator-initialisierte Anfragen, wie beispielsweise Sollpunktänderungen, Abstimmparameter, Zugriff und Änderungen, Alarmbestätigungen und Vorrichtungs-up und -downloads zu bewirken.

Ein Reportverteiler VCR wird für die in eine Warteschlange eingereihte, nicht geplanten, anwenderinitialisierten Kommunikationen von einer bis mehreren Kommunikationen verwendet. Wenn beispielsweise eine Bereichsvorrichtung mit einem Ereignis- oder einem Trendreport ein Durchgangs-Token von einer LAS empfängt, so sendet diese Bereichsvorrichtung ihre Nachricht zu einer "Gruppenadresse", die in der Feldbus-Zugriffs-Sub-Schicht des Kommunikationsstapels dieser Vorrichtung definiert ist. Die Vorrichtungen, die dafür konfiguriert sind, um auf dieses VCR zu horchen, werden den Report empfangen. Der Reportverteilungs-VCR-Typ wird in typischer Weise durch die

Fieldbus-Vorrichtungen verwendet, um Alarmbenachrichtigungen zu Bedienungskonsolen zu senden.

Ein Herausgeber/Benutzer-VCR-Typ wird dazu verwendet, um eine bis viele Kommunikationen zu puffern. Die gepufferten Kommunikationen sind solche, die lediglich die späteste Version der Daten speichern und senden und es überschreiben daher neue Daten vollständig die früheren Daten. Die Funktionsausgangsgrößen umfassen beispielsweise gepufferte Daten. Eine "Veröffentlicher"-Bereichsvorrichtung veröffentlicht oder sendet eine Nachricht unter Verwendung des Veröffentlicher/Benutzer-VCR-Typs an alle die "Benutzer"-Bereichsvorrichtungen an den Fieldbus-Bus, wenn die Veröffentlicher-Vorrichtung eine Zwangsdatennachricht von der LAS oder von einer Benutzer- oder Teilnehmer-Vorrichtung empfängt. Die Veröffentlicher-/Benutzerbeziehungen sind vorbestimmt und sind innerhalb der Fieldbus-Zugriffs-Sub-Schicht des Kommunikationsstapels von jeder Bereichsvorrichtung definiert und gespeichert.

Um richtige Kommunikationsaktivitäten über den Fieldbus-Bus sicherzustellen, sendet jede LAS periodisch eine Zeitverteilungsnachricht an alle Bereichsvorrichtungen, die an ein Segment des Busses angeschlossen sind, was des den empfangenden Vorrichtungen ermöglicht, ihre örtliche Datenverbindungszeit so einzustellen, daß sie miteinander synchronisiert sind. Zwischen diesen Synchronisationsnachrichten wird die Taktzeit unabhängig in jeder Vorrichtung beibehalten, und zwar basierend auf deren eigenem internem Takt. Eine Taktsynchronisation erlaubt es den Bereichsvorrichtungen, die Funktionsblock-Programmausführung mit dem Netzwerk zu synchronisieren.

Wie aus der oben gegebenen Erläuterung des Fieldbus-Kommunikationsprotokolls hervorgeht, erfordert eine Kommunikation mit irgendeiner speziellen Vorrichtung oder einem Funktionsblock, der innerhalb des Fieldbus-Abschnitts des Prozeßregelnetzwerks 10 gelegen ist, das heißt an den Bus 42 angeschlossen ist, detailliertes Wissen darüber, auf welche Weise die Kommunikation innerhalb des Fieldbus-Protokolls im allgemeinen bewirkt wird, als auch eine Betrachtung darüber und ein Wissen davon, wie der Fieldbus-Bus 42 speziell aufgebaut ist und wann Kommunikationen, die diesem zugeordnet sind, geplant sind und erlaubt werden. Dies macht es seinerseits für den Designer der Prozeßregelroutine schwierig, die in dem Kontroller 12 implementiert ist, Kommunikationen mit den Vorrichtungen innerhalb des Fieldbus-Abschnitts des Prozeßregelnetzwerks zu implementieren und zu planen, das heißt den Vorrichtungen, die an den Bus 42 angeschlossen sind. Da darüber hinaus Standard- oder reguläre Kommunikationen zwischen den Funktionsblöcken in einer synchronen Weise über den Bus 42 erfolgen, muß der Kontroller 12 so konfiguriert sein, um all diese Kommunikationen oder Nachrichten oder wenigstens die einen zu empfangen, die er zur Ausführung der Steuer- oder Regelroutine benötigt. Es kann eine entmutigende Aufgabe auf Seiten des Kontrollers 12 werden, den Empfang und die Speicherung von all den Informationen zu koordinieren, die auf dem Bus 42 fließen, und zwar in einer Weise, die effizient durch den Kontroller 12 verwendet werden kann, um die Steuerung des gesamten Prozesses 19 zu bewirken. Um ferner Informationen zu empfangen, die nicht synchron über den Bus 42 gesendet werden, muß der Kontroller 12 eine Anfrage nach dieser Information aussenden, die, wie oben erläutert wurde, asynchron über den Bus 42 gesendet wird. Da es für den Kontroller 12 keinen Weg gibt, zu bestimmen oder zu bewirken, wann die asynchrone Anfrage an die geeignete Vorrichtung übergeben wird (da die Zeitsteuerung diese Anfrage direkt durch die anderen asynchronen Kommunikationen beeinflusst wird, die auf dem Bus 42 stattfinden), kann der Kontroller 12 Informationen

empfangen, die nicht mehr aktuell sind, und zwar zu dem Zeitpunkt, wenn sie den Kontroller 12 erreichen. In ähnlicher Weise hat der Kontroller 12 keine Möglichkeit zu bestimmen, welche Daten es zu einem spezifischen Zeitpunkt sein sollen oder waren, was bei der Steuerung anderer Vorrichtungen innerhalb des Prozesses 19 wichtig sein kann, die nicht an den Bus 42 angeschlossen sind. Darüber hinaus kann es schwierig oder unmöglich für den Kontroller 12 sein, wichtige Informationen zu empfangen, die den Status einer Vorrichtung oder eines Funktionsblockes betreffen, und zwar innerhalb des Fieldbus-Abschnitts des Prozesses 19, wie beispielsweise Alarme, die durch solche Funktionsblöcke erzeugt werden.

Um diese Probleme zu überwinden, kann der Kontroller 12 so ausgelegt sein, um die Schattenfunktionsblöcke zu verwenden, um die Kommunikation mit den Vorrichtungen und den Funktionsblöcken innerhalb des Fieldbus-Abschnitts des Prozeßregelnetzwerks 10 zu implementieren. Spezieller gesagt, kann die Steueroutine innerhalb des Kontrollers 12 direkt mit einem Schattenfunktionsblock innerhalb des Kontrollers 12 kommunizieren, der dann automatisch mit einem zugeordneten aktuellen externen Funktionsblock innerhalb einer Bereichsvorrichtung kommuniziert, die an den Bus 42 angeschlossen ist. Der Schattenfunktionsblock innerhalb des Kontrollers 12 ist so konfiguriert, um den Stand der und die Daten, die dem aktuellen externen Funktionsblock zugeordnet sind, innerhalb einer Vorrichtung nach unten zu spiegeln, die an den Bus 42 angeschlossen ist, so daß es für die Prozeßsteueroutinen innerhalb des Kontrollers 12 so scheint, als ob der tatsächliche externe Funktionsblock zugreifbar ist, ohne vermittels des Fieldbus-Protokolls über den Bus 42 kommunizieren zu müssen. Mit anderen Worten erscheint es für die Prozeßsteueroutine innerhalb des Kontrollers 12 so, als ob der tatsächliche Funktionsblock innerhalb des Prozeßkontrollers 12 gelegen wäre, anstatt unten innerhalb einer externen Bereichsvorrichtung, die an den Bus angeschlossen ist, und die Prozeßsteueroutine verwendet den Schattenfunktionsblock, so wie sie andere Funktionsblöcke innerhalb des Kontrollers 12 verwendet.

Fig. 4 veranschaulicht eine graphische Einzelheit einer Prozeßregelschleife oder eines Moduls 100, der der Prozeßsteueroutine zugeordnet ist und durch diese implementiert ist, und zwar innerhalb des Prozeßkontrollers 12. Um die Steuerung des gesamten Prozesses 19 zu implementieren, kann die Prozeßsteuer- oder -regelroutine innerhalb des Kontrollers 12 viele solcher Schleifen oder Module implementieren, die in irgendeiner gewünschten Weise miteinander verbunden sein können. Die herausgegriffene Prozeßregelschleife 100 enthält eine Repräsentation eines Aspektes (Flusses) des Prozesses 101, der durch eine Anzahl von Kontrollerblöcken oder Einheiten gesteuert wird, speziell durch einen PID-Block 102, der an einen AO-Block 104, einen ersten AI-Block 106 und einen zweiten AI-Block 108 gekoppelt ist. Jeder der Blöcke 102, 104 und 106 ist eine graphische Einzeldarstellung einer Regelsubroutine (oder eines Objektes innerhalb einer objektorientierten Programmierung), die der Prozeßregelroutine zugeordnet ist, die in dem Kontroller 12 gespeichert ist, konfiguriert gemäß einem Kontrollerprotokoll, welches dem Kontroller 12 zugeordnet ist und dazu verwendet wird, um einen Abschnitt einer Gesamtregelstrategie in bezug auf den Prozeß 101 zu implementieren. Jeder der Blöcke der Prozeßregelroutine 100 enthält Eingänge und Ausgänge (durch Eingänge auf der linken und rechten Seite der Blöcke jeweils veranschaulicht) und kann einen Steuer- oder Regelalgorithmus enthalten, der eine gewisse Steuer- oder Regelfunktion durchführt. Verbindungen untereinander oder Verbindungen zwischen

den Funktionsblöcken sind als Leitungen oder Linien dargestellt, die von einem Ausgang von einem Block zu einem Eingang eines anderen Blocks verlaufen. Diese Verbindungen geben die Art wieder, in welcher die Kommunikation zwischen den einzelnen Blöcken innerhalb der Steuerroutine oder Modul implementiert wird, um eine Prozeßregelschleife gemäß einem Kontrollerprotokoll auszuführen. Somit veranschaulicht die Einzelheit von Fig. 4 nicht nur die Elemente der Regelschleife, die ausgeführt werden, sondern auch die Art, in welcher die Prozeßregelroutine innerhalb des Kontrollers 12 ausgelegt ist, um diese Schleife zu implementieren. Die Prozeßregelroutine kann automatisch geändert oder erneut konfiguriert werden, und zwar durch Bewegen der Verbindungen zwischen den Blöcken, durch Addieren oder Weglassen von Blöcken davon usw., wie dies durch den Delta V-Prozeßkontroller ausgeführt wird.

Der PID-Funktionsblock 102, der in Fig. 4 veranschaulicht ist, enthält einen Algorithmus (durch den Kontroller 12 implementiert), der beispielsweise eine Proportional/Integral/Differential-Regelberechnung ausführt, basierend auf den Eingangsgrößen, die er von den AI-Blöcken 106 und 108 und dem AO-Block 104 empfängt, und der ein Ausgangssteuersignal für den AO-Block 104 erzeugt, welcher seinerseits eine Vorrichtung (28 beispielsweise ein Ventil) innerhalb des Prozesses 101 veranlaßt, eine Funktion durchzuführen, wie beispielsweise bewirken, daß ein Ventil bewegt wird, um die Strömung eines Mediums zu erhöhen. Der AO-Block 104 kann einem tatsächlichen Ventil zugeordnet sein und dieses steuern, z. B. das Ventil 28 von Fig. 1, und zwar über die IO-Vorrichtung 20 und die 4-20-mA-Kommunikationsleitung 38. Der AO-Block 104 empfängt eine Messung der tatsächlichen Position des Ventils und liefert diese Messung als eine Rückkopplungsgröße zu dem PID-Funktionsblock 102 über die Verbindung 110. Darüber hinaus empfängt der PID-Funktionsblock 102 Eingangsgrößen von dem AI-Block 106, der beispielsweise einem Sensor, wie einem Temperatursensor, zugeordnet ist, der innerhalb des Prozesses 19 gelegen ist. Dieser Sensor kann beispielsweise der Sensor 34 von Fig. 1 sein, in welchem Fall der AI-Block 106 die Sensormeßgrößen über die I/O-Vorrichtung 22 empfängt, und zwar unter Verwendung von Standardnachrichtenübertragungen. Solch eine Verbindung ist in Fig. 4 durch die Verbindung zwischen dem Ausgang, der mit "Fluß" in dem Prozeß 101 bezeichnet ist, und dem Eingang des AI-Blocks 106, der mit "simulate_in" bezeichnet ist, veranschaulicht. Die Verarbeitung und die Steuerung der Information, die dem PID-Funktionsblock 102, dem AO-Block 104 und dem AI-Block 106 zugeordnet ist, wird innerhalb des Kontrollers 12 durchgeführt.

Allgemein gesagt, ist innerhalb des Delta V-Steuersystems, das heißt Verwendung des Delta V-Kontrollerkonfigurationsprotokolls jeder der Blöcke 102, 104 und 106 spezifisch konfiguriert, um all die Informationen und Daten zu unterstützen, die den ähnlichen Funktionsblöcken in dem Fieldbus-Protokoll zugeordnet sind und somit für alle Absichten und Zwecke, und ist sehr ähnlich einem Funktionsblock innerhalb des Fieldbus-Protokolls mit der Ausnahme, daß die Steuerfunktionen oder Regelfunktionen durch den zentralen Prozessor 12 durchgeführt werden und daß die Informationen von speziellen Bereichsvorrichtungen über Standardkommunikationsleitungen von dem Prozeßkontroller 12 empfangen und übergeben werden. Somit sind die Abschnitte der Steuer- oder Regelroutine, die in Fig. 4 herausgegriffen ist, welche die Blöcke 102, 104 und 106 enthält, momentan in der Delta V-Umgebung vorgesehen und sind bekannt.

Um die Fieldbus-Integration innerhalb des Kontrollers 12 zu unterstützen, kann die folgende Annäherung versucht

werden, und zwar unter Verwendung eines Delta V-Steuersystems (oder eines anderen zentralisierten Steuersystems), bei dem der Basissatz an Funktionsblöcken, die in dem Steuersystem oder Regelsystem verwendet werden, ähnlich denjenigen ist, die durch das Fieldbus-Protokoll definiert sind. Die externen Fieldbus- oder herstellerspezifischen Funktionsblöcke sind individuell zugeordnet, um in Verbindung mit dem Steuer- oder Regelsystem (oder einem Teil des Steuer- oder Regelsystems) ein Programm auszuführen und die Funktionsblöcke der Fieldbus-Vorrichtungen sind in dem Steuer- oder Regelsystem als Schattenfunktionsblöcke wiedergegeben oder reflektiert, welche interne und externe Referenzen unterstützen, so als ob die externen Funktionsblöcke in dem Steuer- oder Regelsystem vorhanden wären. Das Steuer- oder Regelsystem erneuert automatisch die dynamischen und die statischen Parameter des Schattenfunktionsblocks und leitet Änderungsanfragen zu den geeigneten externen Funktionsblöcken unter Verwendung der Schattenfunktionsblöcke. Alarmsignale, die in den externen Vorrichtungen (oder Funktionsblöcken) detektiert werden, werden in den Schattenfunktionsblöcken reflektiert und werden in der Alarmverarbeitung des Steuer- oder Regelsystems integriert. Natürlich kommuniziert der Schattenfunktionsblock mit den externen Funktionsblöcken unter Verwendung des Kontrollerprotokolls, welches den externen Funktionsblöcken zugeordnet ist, welches verschieden von dem Kontrollerkonfigurationsprotokoll sein kann und typischerweise verschieden ist, welches von dem Kontroller verwendet wird, um die Kommunikationen zwischen den Funktionsblöcken intern zu dem Kontroller zu implementieren. Auch können Verbindungen zwischen internen und externen Funktionsblöcken durch den Anwender in einer Weise bestimmt werden, die davon abhängt, wo der Funktionsblock vorhanden ist. Als ein Ergebnis erscheinen während der Steuerdefinition und der Online-Diagnose die Funktionsblöcke als die gleichen, ob sie nun interne (in dem Steuer- oder Regelsystem vorhanden) oder externe (in einer Fieldbus-Vorrichtung vorhanden) sind oder nicht.

Somit ist gemäß der vorliegenden Erfindung der AI-Block 108, der in Fig. 4 so dargestellt ist, daß er ähnlich dem AI-Block 106 ist, tatsächlich ein Schattenfunktionsblock, der so konfiguriert ist, daß er mit einem externen Funktionsblock 112 kommuniziert, der beispielsweise innerhalb des Sensors 48 gelegen ist, der an den Fieldbus 42 von Fig. 1 angeschlossen ist. Der Schattenfunktionsblock 108 liefert gemessene oder andere Signale, die dem externen Funktionsblock 112 zugeordnet sind, an den PID-Block 102 über die Verbindungen, die dazwischen erstellt wurden. Um anzuzeigen, daß der Block 108 ein Schattenfunktionsblock ist, der einem externen Funktionsblock 112 zugeordnet ist und mit diesem kommuniziert, unter Verwendung des Kommunikationsprotokolls des externen Funktionsblockes 112, ist der Block 108 so dargestellt, daß er eine strichlierte Linie besitzt, die an den AI-Funktionsblock 112 innerhalb der Fieldbus-Vorrichtung 48 angehängt ist. Jedoch kann der Schattenfunktionsblock 108 so dargestellt sein, daß er eine Vorrichtungsetikette (device tag) und/oder einen Blocknamen am Boden desselben besitzt, oder kann in irgendeiner anderen gewünschten Weise dargestellt sein, wobei darauf hingewiesen sei, daß die Art, in welcher der Schattenfunktionsblock für einen Anwender als Einzelheit dargestellt ist, nicht kritisch ist, und zwar in Verbindung mit dem Betrieb des Schattenfunktionsblocks. Ferner werden die Eingangsgrößen und Ausgangsgrößen, die dem AI-Funktionsblock 112 zugeordnet sind, innerhalb des Fieldbus-Netzwerks gemäß dem Weg übergeben, in welchem das Netzwerk konfiguriert worden ist.

Während der Schattenfunktionsblock 108 auf alle oder

die meisten der Informationen zugreift oder spiegelt, die dem tatsächlichen Funktionsblock 112 zugeordnet sind und/oder durch diesen erzeugt werden, erfolgt irgendeine Verarbeitung, die in bezug auf die AI-Funktionsblock 112 stattfindet (oder in bezug auf irgendeinen anderen Funktionsblock, für den ein Schattenfunktionsblock existiert) in dieser Weise unten in der externen Vorrichtung, nicht in dem Schattenfunktionsblock 108 oder selbst in dem Kontroller 12. Als ein Ergebnis kann der Schattenfunktionsblock 108 so betrachtet werden, als ob er eine Röhre von Informationen zwischen dem PID-Block 102 (oder irgendeinem anderen Block innerhalb des zentralisierten Kontrollers 12) und dem tatsächlichen Funktionsblock 112 bilden würde. Der Schattenfunktionsblock 108 ist nicht ein Funktionsblock, der vollständig durch den Kontroller 12 betreibbar ist, in dem Sinn, daß der AI-Block 106 und der PID-Block 102 durch den Kontroller 12 betreibbar sind, da der Schattenfunktionsblock 108 lediglich die Informationen innerhalb des tatsächlichen externen Funktionsblocks 112 spiegelt oder ansonsten eine Nachrichtenübertragung zwischen dem Kontroller 12 und dem tatsächlichen externen Funktionsblock 112 vorsieht. Nichtsdestoweniger kann durch eine geeignete Operation des Schattenfunktionsblockes der Kontroller 12 den tatsächlichen Funktionsblock 112 steuern und mit diesem kommunizieren, so als ob dieser vollständig in dem Kontroller 12 implementiert wäre. Indem beispielsweise eine Kommunikation mit dem Schattenfunktionsblock 108 durchgeführt wird, unter Verwendung des Kontrollerkonfigurationsprotokolls, kann die Prozeßsteuer- oder -regelroutine innerhalb des Kontrollers 12 auf den neuesten Stand gebrachte Informationen von dem Funktionsblock 112 empfangen und kann Befehle zu dem Funktionsblock 112 so schnell wie möglich senden, ohne sich darum kümmern zu müssen, wo der tatsächliche Funktionsblock 112 gelegen ist oder auf welche Weise Kommunikationen mit dem tatsächlichen Funktionsblock 112 bewirkt werden. Statt dessen erfolgen die Kommunikationen zwischen dem tatsächlichen Funktionsblock 112 und dem Schattenfunktionsblock 108 automatisch ohne Eingreifen durch die Prozeßsteueroutine, die lediglich die Schritte ausführen muß, die sie in typischer Weise ausführt, um die Kommunikationen zwischen den Steuer- oder Funktionsblöcken zu implementieren, die innerhalb des Kontrollers 12 gelegen sind. In dieser Weise ermöglicht es der Schattenfunktionsblock einem Kontroller, der Funktionsblöcke innerhalb des Kontrollers implementiert, Funktionsblöcke zu verwenden und mit zu integrieren, die unterschiedliche Kommunikations- oder Konfigurationsprotokolle verwenden und die nicht innerhalb des Kontrollers gelegen sind, sondern die statt dessen in einer externen Vorrichtung gelegen und in diese implementiert sind, wie beispielsweise einer mikroprozessor-unterstützten Bereichsvorrichtung, die in einer Prozeßumgebung angeordnet ist. Diese hinzugefügte Fähigkeit wird von dem Kontroller 12 in transparenter Weise erreicht, so daß dann, sobald der Schattenfunktionsblock in dem Kontroller 12 aufgebaut ist, die Steuer- oder Regelroutine nicht nachzusuchen braucht, wo der tatsächliche Funktionsblock gelegen ist, um komplexe Kommunikationen oder Datenbasismanipulationen durchzuführen, um den externen Funktionsblock 112 zu verwenden.

Wie hervorgeht, speichert der Schattenfunktionsblock 108 die Eingangsgrößen und Ausgangsgrößen der parametrischen Updates, die dem AI-Funktionsblock 112 in einer Datenbasis in dem Kontroller 12 zugeordnet sind, und zwar in irgendeinem gewünschten Format, speichert jedoch in bevorzugter Weise diese in dem gleichen oder einem ähnlichen Format wie die Informationen, die den Steuerblöcken zugeordnet sind, die durch den Kontroller implementiert sind,

das heißt unter Verwendung des Kontrollerkonfigurationsprotokolls. Dies macht die Kommunikation mit dem Schattenfunktionsblock 108 für den Kontroller 12 transparent, das heißt die Prozeßsteuer- oder -regelroutine 100 schafft eine Kommunikation in bezug auf den Schattenfunktionsblock 108 (und somit mit dem tatsächlichen Funktionsblock 112) über die Verbindung in der gleichen Weise wie sie eine Kommunikation in bezug auf die anderen Funktionsblöcke 104 und 106 erzeugt. Während es wünschenswert ist, daß die Funktionsblöcke innerhalb des Kontrollers 12 logisch konfiguriert sind, um all die Informationen (oder die meisten) zu unterstützen, die durch das dezentralisierte oder externe Prozeßregelnetzwerk unterstützt werden (wie dies bei dem Delta V-Kontroller und dem Fieldbus-Kommunikationsprotokoll der Fall ist), so ist diese Konfiguration nicht erforderlich. Tatsächlich kann irgendein Kontrolleraufbau unter Verwendung des Schattenfunktionsblockes unterstützt werden, der hier offenbart ist, indem die Daten, die in dem externen Funktionsblock unterstützt werden und in diesem verfügbar sind, in den Daten abgebildet werden, die in der Kontrolleroutine verwendet und verfügbar sind.

Allgemein gesagt, um den Betrieb des Schattenfunktionsblockes 108 in dem Fieldbus-Netzwerk zu bewirken, wird der tatsächliche Funktionsblock 112 so konfiguriert, um die verketteten Daten zu veröffentlichen und an den Prozeßkontroller 12 (das heißt die Daten, die mit einem anderen Block innerhalb des Kontrollers 12 über eines der Verbindungsglieder, die in Fig. 4 veranschaulicht sind, verbunden sind) oder zu veröffentlichen, und zwar über die Schnittstellenkarte 40, unter Verwendung des Veröffentlichers/Teilnehmer-VCR's (das heißt synchronen Kommunikationen) in dem Fieldbus-Kommunikationsprotokoll. Andere nicht verkettete Daten, die dem tatsächlichen Funktionsblock 112 zugeordnet sind, werden an die Schnittstellenkarte 40 unter Verwendung von asynchronen Kommunikationen bzw. Nachrichtenübertragungen übergeben, unter Verwendung von beispielsweise den Betrachtungsobjekt- oder Warnobjekt-funktionen innerhalb des Fieldbus-Protokolls, was in der Modulabstrakte des Steuermoduls innerhalb des Kontrollers 12 stattfindet. Es werden daher in typischer Weise verkettete Daten zu dem Kontroller 12 von dem Funktionsblock 112 in einer sehr viel schnelleren Rate gesendet als andere Daten (weniger zeitsensitive Daten). Umgekehrt werden verkettete Daten, die durch einen Block innerhalb des Kontrollers 12 zu dem Schattenfunktionsblock 108 gesendet werden, unmittelbar an die Schnittstellenvorrichtung 40 weitergeleitet, wo sie auf dem Fieldbus-Bus 42 unter Verwendung von synchronen Standardnachrichtenübertragungen veröffentlicht werden.

Die Informationen, die von dem tatsächlichen Funktionsblock 112 gesendet werden, werden automatisch in den Schattenfunktionsblock 108 gesetzt und stehen somit für die Steuer- oder Regelroutine in dem Kontroller 12 zu jeder Zeit zur Verfügung. Um diese Operation zu bewirken, nimmt die Schnittstellenkarte 40 (die Teil eines Kommunikationsports des Schattenfunktionsblockes sein kann) teil an den veröffentlichten Verbindungsparameterdaten, die durch den Funktionsblock 112 erzeugt wurden, und liefert diese Informationen zu dem Prozeßkontroller 12 in der Rate, die durch die Ausführungsrate des Funktionsblockes innerhalb des Makrozyklusses des Fieldbus-Segments festgelegt ist, an welches der externe Funktionsblock 112 angeschlossen ist. In ähnlicher Weise erhält die Schnittstellenkarte 40 (die typischerweise das LAS des Fieldbus-Segments ist) die Betrachtungsobjekte und/oder Warnobjekte des tatsächlichen Funktionsblocks in einer Rate, die durch die Abstrakte des Kontrollermoduls festgelegt ist, in welchem der Schattenfunktionsblock 108 vorhanden ist, das heißt in der Rate, in

welcher die Steuer- oder Regelroutine 100 in dem Controller 12 die Blöcke implementiert, die dieser zugeordnet sind. Wie dies bekannt ist, enthält das Fieldbus-Protokoll vier Typen von Betrachtungsobjekten, die die normalerweise dynamischen (Betrachtungsobjekt 1), normalerweise statischen (Betrachtungsobjekt 2), alle dynamischen (Betrachtungsobjekt 3) und alle statischen (Betrachtungsobjekt 4) Variablen speichern. Die Schnittstellenkarte 40 ist so konfiguriert, um automatisch auf einer periodischen Grundlage oder durch Aussenden einer Anfrage nach solchen Informationen das dynamische Betrachtungsobjekt 3 zu empfangen, welches die Werte für alle die dynamischen Variablen enthält, die dem tatsächlichen Funktionsblock 112 zugeordnet sind. Nach Empfangen dieser Betrachtungsobjektinformationen speichert die Schnittstellenkarte 40 die Informationen in einer Datenbasis oder Datenbank, die durch den Schattenfunktionsblock zugreifbar ist, und der Schattenfunktionsblock 108 erneuert die Variablen, die sich dadurch geändert haben, wodurch diese variablen Werte für die Prozeßsteuer- oder -regelroutine innerhalb des Controllers 12 verfügbar werden. Wenn eine der dynamischen Variablen (die statische Revisionsvariable) eine Änderung an einer statischen Variablen anzeigt, kann der Schattenfunktionsblock oder die Software innerhalb der Schnittstellenkarte 40 anfragen, daß das Betrachtungsobjekt 4 (alle statischen Variablen) zu dem Schattenfunktionsblock 108 gesendet werden, um die statischen Variablen auf den neuesten Stand zu bringen. Bei der bevorzugten Ausführungsform empfängt die H1-Schnittstellenkarte 40 fortlaufend das Betrachtungsobjekt 3 in der Modulabtastrate des Steuermoduls 100 innerhalb des Controllers 12.

Wie oben dargelegt ist, kann der Controller 12 (oder irgendein Funktionsblock desselben) Befehle zu dem aktuellen Funktionsblock 112 innerhalb des Fieldbus-Netzwerks über den Schattenfunktionsblock 108 einfach dadurch senden, indem ein Parameter innerhalb des Schattenfunktionsblockes 108 geändert wird. Diese Parameteränderung wird automatisch nach unten zu dem AI-Funktionsblock 112 über einen Ausgang des Schattenfunktionsblockes 108 gesendet, wo er dazu verwendet wird, um die Konfiguration des AI-Funktionsblockes 112 zu ändern. Während die Datenänderung oder der Einschreibebefehl in dem tatsächlichen Funktionsblock 112 nicht zu exakt der gleichen Zeit durchgeführt wird, zu der dieser durch den Schattenfunktionsblock 108 empfangen wird, und zwar vom Standpunkt des Controllers 12 aus, erfolgt diese Änderung sehr schnell und wird in den Schattenfunktionsblock 108 zurück reflektiert, wenn die Änderung tatsächlich vorgenommen wurde. Diese Betriebsweise ermöglicht es dem Controller 12 und speziell dem PID-Funktionsblock 102, innerhalb des Controllers 12 eine Änderung zu bewirken, indem lediglich diese Änderung in eine Speicherstelle eingeschrieben wird, die dem Schattenfunktionsblock 108 zugeordnet ist und indem danach die Kommunikation automatisch vorgenommen wird, ohne daß dabei irgendwelche speziellen Kommunikationsaktivitäten ausgeführt werden müssen, die erforderlich sind, um Daten in das Fieldbus-Netzwerk hinein zu bekommen und heraus zu bekommen.

Neben den Eingangs- und Ausgangsparameterdaten können andere Typen von Daten, wie beispielsweise Modus- und Statusdaten, zwischen einem Controllerfunktionsblock (das heißt einem internen Funktionsblock) und dem Schattenfunktionsblock 108 übertragen werden als auch zwischen dem Schattenfunktionsblock 108 und dem tatsächlichen externen Funktionsblock 112. Natürlich wird diese Information mit den Betrachtungsobjekt- oder Warnobjektinformationen von dem externen Funktionsblock 112 zu dem Schattenfunktionsblock 108 gesendet. In ähnlicher Weise können

solche Modus- und Statusdaten an den Schattenfunktionsblock 108 von einem internen Funktionsblock innerhalb des Controllers 12 übermittelt werden und es können diese Daten dann an den externen Funktionsblock 112 für die Verwendung geliefert werden. Der Statusparameter eines Funktionsblockes zeigt in typischer Weise die Qualität der Messung oder der Daten an, die durch den Funktionsblock geliefert werden, und kann Gründe liefern, warum eine schlechte Qualität existiert. Er kann auch eine Grenze anzeigen, wie beispielsweise eine hohe oder niedrige Grenze, welche die Daten erreichen können. Der Modusparameter zeigt in typischer Weise an, welcher Modus des Funktionsblocks eingeschaltet ist, welcher beispielsweise ein Handmodus, ein Automatikmodus oder ein Kaskadenmodus sein kann, wie dies durch das Fieldbus-Protokoll festgelegt ist.

Darüber hinaus wird eine Alarmdetektion basierend auf Alarmsignalen, die in dem tatsächlichen externen Funktionsblock erzeugt werden, vorgesehen, da Ereignisse und Berichte durch den externen Funktionsblock automatisch als dynamische Parameter erzeugt werden, die dann durch die Betrachtungs- oder Warnobjekte dem Schattenfunktionsblock 108 angeboten werden. Als ein Ergebnis kann eine Alarminformation, die den externen Funktionsblock betrifft, von dem Controller 12 betrachtet und verwendet werden.

Obwohl natürlich der Block 108 hier als Schattenfunktionsblock beschrieben wurde, können irgendwelche Blöcke innerhalb der Fig. 4 ebenfalls oder alternativ Schattenfunktionsblöcke sein.

Die Vorteile, die mit der Verwendung eines Schattenfunktionsblockes verbunden sind, wie beispielsweise dem Schattenfunktionsblock 108, der in Fig. 4 veranschaulicht ist, bestehen darin, daß dieser es dem Prozeßregeldesigner ermöglicht, die Steuerung oder Regelung innerhalb eines zentralisierten Prozeßcontrollers 12 zu implementieren unter Verwendung von externen Funktionsblöcken, das heißt Funktionsblöcken, die tatsächlich innerhalb einer verschiedenen Vorrichtung implementiert sind und die unterschiedlichen Kommunikationsprotokollen unterworfen sind. In Verbindung mit dem Schattenfunktionsblock braucht ein Designer oder Konstrukteur sich nicht darum zu kümmern, daß der externe Funktionsblock einem unterschiedlichen Kommunikations- oder Steuer- oder Regelprotokoll zugeordnet ist oder in einer unterschiedlichen Vorrichtung gelegen ist, da die Nachrichtenübermittlung zwischen dem Schattenfunktionsblock und dem externen Funktionsblock automatisch erfolgt und auch transparent für die Steuer- oder Regelroutine. Wenn ferner einmal ein Schattenfunktionsblock implementiert ist und läuft, ist es einfach ein Modell darüber zu erstellen, was sich innerhalb des gesamten Prozeßregelsystems ereignet, und zwar ohne sich darum zu kümmern, ob Aktionen innerhalb einer Fieldbus- oder anderen externen Vorrichtung auftreten oder innerhalb des Controllers auftreten, da der Schattenfunktionsblock für den Controller und den Anwender bewirkt, daß es so scheint, daß der tatsächliche Funktionsblock in dem Controller implementiert ist, obwohl dies real nicht der Fall ist.

Wenn ein gemeinsamer Funktionsblocksatz und Schattenfunktionsblöcke in dem Steuer- oder Regelsystem verwendet werden, um Funktionsblöcke wiederzugeben, die externen Vorrichtungen zugeordnet sind, kann die Steuer- oder Regelstrategie zu Beginn ohne das Wissen ausgelegt werden, ob ein bestimmter Funktionsblock dieser Strategie in dem Steuer- oder Regelsystem vorhanden sein wird oder in einer externen Vorrichtung vorhanden sein wird, und es können Anwenderanwendungen auf die Schattenfunktionsblöcke zugreifen (welche die externen Funktionsblöcke wiedergeben oder als Modell darstellen), und zwar in exakt der gleichen Weise wie sie auf die Steuer- oder Regelsystem-

funktionsblöcke zugreifen. Auch werden Alarme, die durch die externe Vorrichtung detektiert werden, voll in die Steuersystemalarmverarbeitung durch den Schattenblock integriert, mit der Möglichkeit, daß diese Alarme in exakt der gleichen Weise für die externen Funktionsblöcke erscheinen wie für Funktionsblöcke, die innerhalb des Steuer- oder Regelsystems vorhanden sind.

Die Konfiguration, Implementierung und Verwendung eines Schattenfunktionsblockes wird nunmehr in Einzelheiten unter Hinweis auf die Fig. 5-13 beschrieben. Fig. 5 veranschaulicht den physikalischen Aufbau eines Abschnitts des Prozeßregelnetzwerks von Fig. 1 mehr in Einzelheiten. Die Anwenderworkstation 150, die aus irgendeinem der PCs 14 von Fig. 1 bestehen kann, ist kommunikativ mit dem Controller 12 verbunden, der seinerseits über die Schnittstellenkarte 40 mit der Bereichsvorrichtung 48 verbunden ist, welche den Funktionsblock 112 darin enthält. Wie in Fig. 5 veranschaulicht ist, besitzt der Controller 12 einen oberen Abschnitt 152, in welchem die Steueroutine 100 (enthaltend den Schattenfunktionsblock 108) implementiert ist, und einen unteren Abschnitt, der eine Datenbasis oder Datenbank 156 enthält, um Eingangs- und Ausgangsinformationen zu speichern, die von der Schnittstellenkarte 40 als auch von anderen I/O-Karten empfangen werden. Allgemein gesagt, ist die Schnittstellenkarte 40 so konfiguriert, um automatisch verkettete Daten zu empfangen, die durch den Funktionsblock 112 veröffentlicht werden, und zwar über den Veröffentlichungs/Teilnehmer-VCR (in der Veröffentlichungsrate, die durch den Makrozyklus des Fieldbus-Busses 42 festgelegt ist), und um diese Daten in der unteren Datenbank 156 zu speichern, wo sie dem Schattenfunktionsblock 108 in der Abtastrate des Steuermodus 100 innerhalb des Controllers 12 angeboten werden. In ähnlicher Weise ist die Schnittstellenkarte 40 so konfiguriert, um die verketteten Daten, die durch einen Funktionsblock innerhalb des Controllers 12 erzeugt werden, auf dem Fieldbus-Bus 40 zu veröffentlichen, unter Verwendung synchroner Nachrichtenübermittlungen innerhalb des Fieldbus-Netzwerks. Auch ist die Schnittstellenkarte 40 so konfiguriert, um periodisch anzufragen nach (unter Verwendung asynchroner Nachrichtenübertragungen) Betrachtungs- und/oder Alarmdaten von dem Funktionsblock 112 und um diese Informationen in der unteren Datenbank 156 zu speichern, um sie an den Schattenfunktionsblock 108 in der Abtastrate des Controllermoduls 100 zu liefern.

Die Schnittstellenkarte 40 und/oder die Datenbasis 156 kann einen Kommunikationsport des Schattenfunktionsblockes 108 umfassen, kann Teil von diesem sein oder kann durch diesen gesteuert sein, der Kommunikationen zwischen dem Schattenfunktionsblock 108 und dem externen Funktionsblock 112 implementiert. Der Kommunikationsport enthält einen Eingang, der von dem externen Funktionsblock 112 Daten empfängt, und zwar unter Verwendung des Kommunikationsprotokolls des externen Funktionsblocks 112, und enthält einen Ausgang, der mit dem externen Funktionsblock 112 in Verbindung steht, um Daten (inklusive Schreibbefehlen) an den externen Funktionsblock 112 unter Verwendung des Kommunikationsprotokolls des externen Funktionsblocks 112 zu senden. Natürlich kann der Kommunikationsport des Schattenfunktionsblocks 108 in der Software und/oder einer anderen Hardware in dem Controller zusätzlich oder in Verbindung mit der Schnittstellenkarte 40 und der Datenbank 156 implementiert sein.

Um eine Steuer- oder Regelroutine unter Verwendung eines Schattenfunktionsblockes zu konfigurieren, kann ein Anwender bei der Workstation 150 irgendwelche Standardwerkzeuge verwenden, wie beispielsweise solche, die durch das Delta V-Steuer- oder -Regelsystem geschaffen werden,

um zu Beginn das Prozeßregelsystem zu konfigurieren. Im allgemeinen ermöglichen es die DeltaV-Konfigurationswerkzeuge einem Anwender, Blöcke darzustellen, zu konfigurieren und miteinander zu verbinden, wie beispielsweise diejenigen, die in Fig. 4 veranschaulicht sind, um eine oder mehrere Regelschleifen zu implementieren oder zu konstruieren oder Module, die der Steuer- oder Regelroutine zugeordnet sind. Zum Zwecke der Erläuterung kann die Steueroutine zum Steuern eines Prozesses irgendeine Anzahl von Modulen enthalten, von denen jeder irgendeine Anzahl von gewünschten Blöcken enthalten kann, die eine Anzahl von Regelschleifen implementieren. Während der Konfiguration oder der Konstruktion eines Prozeßregelmoduls kann ein Anwender die Verwendung eines Funktionsblockes auswählen, der innerhalb einer Vorrichtung extern zum Controller 12 gelegen ist (wie beispielsweise einen Funktionsblock innerhalb einer der Fieldbus-Vorrichtungen des Prozeßregelsystems). In diesem Fall kann das Konfigurationswerkzeug, welches dem Controller 12 zugeordnet ist, so konstruiert sein, um den Anwender zu fragen, die physikalischen der Verbindungen der Vorrichtung zu dem Controller 12 zu spezifizieren und andere Vorrichtungs- und Funktionsblockkonfigurationsinformationen zu spezifizieren, die für die anfängliche Konfiguration der Vorrichtung und/oder des Funktionsblockes innerhalb der Vorrichtung gemäß dem Kommunikationsprotokoll dieser Vorrichtung erforderlich sind (z. B. das Fieldbus-Kommunikationsprotokoll). Der Anwender kann dann aufgefordert werden, diese Informationen in irgendeiner gewünschten Weise zu liefern, wie beispielsweise über die Verwendung von Dialogkästchen. Natürlich hängt die exakte Information, die benötigt wird, von dem Typ der Vorrichtung und des Funktionsblockes ab, der spezifiziert wird, wie dies von einem Fachmann in Verbindung mit dem Vorrichtungsprotokoll, welches verwendet wird, wie beispielsweise dem Fieldbus-Protokoll, zu erkennen ist.

Ein Weg, um zu spezifizieren, daß ein Funktionsblock durch eine bestimmte externe Vorrichtung (wie beispielsweise eine Fieldbus-Vorrichtung) zu implementieren ist, besteht darin, den Anwender dazu zu bringen, den Funktionsblock, der innerhalb einer externen Vorrichtung gelegen ist, zu spezifizieren, unter Verwendung eines Browsers (oder einer anderen Software) oder einer Liste oder durch Herausgreifen der externen Vorrichtungen, die an den Controller angeschlossen sind, und dann Auswählen von einer der aufgelisteten externen Vorrichtungen als die Vorrichtung, in welcher der Funktionsblock zu implementieren ist. Ein anderer Weg besteht darin, den Funktionsblock auf dem Bildschirm auszuwählen und dann diesen Funktionsblock auf eine Einzelheit eines Blocks in einer externen Vorrichtung durch Drag and Drop zu bewegen. Irgendeine dieser oder irgendeine andere gewünschte Aktion kann dem Konfigurationswerkzeug mitteilen, daß der Funktionsblock, der implementiert werden soll, innerhalb der externen Vorrichtung liegt, und kann das Konfigurationswerkzeug veranlassen, den Anwender nach der spezifischen Vorrichtung und den Funktionsblockkonfigurationsinformationen zu fragen, die erforderlich sind, um den speziellen externen Funktionsblock zu konfigurieren und auf diesen zuzugreifen. Beide diese Verfahren ermöglichen es einem Anwender, einen Funktionsblock in einer externen Vorrichtung aus einer Liste von externen Vorrichtungen für die Ausführung eines Programms auszuwählen.

Wenn der Anwender spezifiziert, daß ein externer Funktionsblock zu verwenden ist (das heißt extern vom Controller 12), so erstellt die Konfigurationsroutine dann einen Schattenfunktionsblock in dem Controller und unternimmt Schritte, um die Fieldbus-Vorrichtung und den Funktions-

block innerhalb der Vorrichtung zu konfigurieren, wie dies im folgenden beschrieben wird. Natürlich erlaubt es das Konfigurationswerkzeug in bevorzugter Weise dem Anwender, alle die Blöcke, die zu verwenden sind, zu spezifizieren, ebenso die Örtlichkeiten oder Lagen solcher Blöcke (das heißt, ob irgendein Block in externen Vorrichtungen gelegen ist) und auch die Verbindungen zwischen den Blöcken, bevor die Schattenblöcke implementiert werden und die externen (z. B. Fieldbus-)Vorrichtungen initialisiert werden. Dies ermöglicht es, die Konfiguration des externen Netzwerks einmal auszuführen, nach dem die Örtlichkeit oder Lage von allen den Blöcken und auch die Natur von allen Verbindungen zwischen den Funktionsblöcken spezifiziert worden ist.

In Verbindung mit der Befähigung des Kontrollers, die Parameterdaten für die Funktionsblöcke innerhalb der externen Vorrichtungen einzusehen oder Zugriff auf diese zu haben (über den Schattenfunktionsblock), kann die Schnittstellenkarte 40 auch Vorrichtungs- und Segmentstatusinformationen an den Controller für Diagnosezwecke liefern. Die Schnittstellenkarte 40 kann beispielsweise eine Liste empfangen, die eine Identifikation der Vorrichtungen enthält, die angenommenenmaßen an einem bestimmten Abschnitt oder Segment des Fieldbus-Netzwerks angehängt oder angeschlossen sind, und sachdienliche Informationen über jede der identifizierten Vorrichtungen enthalten. Die Schnittstellenkarte 40 kann dann periodisch Informationen (über synchrone oder asynchrone Nachrichtenverbindungen) erhalten, welche die Vorrichtungen betreffen, die tatsächlich an das Fieldbus-Segment angeschlossen sind oder die das Segment selbst betreffen, und kann diese Informationen mit den Informationen innerhalb der empfangenen Liste vergleichen. Auf diese Weise kann die Schnittstellenkarte 40 bestimmen, ob die Bereichsvorrichtungen fehlen oder nicht an das Fieldbus-Netzwerk angeschlossen sind, ob falsche Bereichsvorrichtungen an das Fieldbus-Netzwerk angeschlossen sind, ob eine Bereichsvorrichtung sich dem Service entzieht, usw. In ähnlicher Weise kann die Schnittstellenkarte 40 bestimmen, ob Segmentwertprobleme auftreten, so als ob keine Nachrichtenverbindung über den Fieldbus-Bus existiert. Wenn es gewünscht wird, kann die Schnittstellenkarte 40 diese Vorrichtungs- und Segmentstatusdaten an den Controller (oder andere Vorrichtung) für Diagnosezwecke senden.

Ferner kann die Schnittstellenkarte 40 die Kommunikation und den Zeitsteuerstatus eines Fieldbus-Segments für Diagnosezwecke überwachen. Insbesondere kann die Schnittstellenkarte 40 automatisch an den Daten teilhaben, die durch die Bereichsvorrichtungen auf dem Fieldbus-Bus veröffentlicht werden, und kann die empfangenen Daten analysieren, um beispielsweise Zeitsteuerprobleme oder andere Probleme auf dem Fieldbus-Bus zu ermitteln. Wenn es gewünscht wird, kann die Schnittstellenkarte 40 die ankommenden Daten zeitlich prägen und speichern und dann Statistiken aufbewahren, die jeden Funktionsblock oder Vorrichtung betreffen, der oder die an den Bus angeschlossen ist und/oder Statistiken aufbewahren, die das Segment des Busses betreffen. Beispielsweise kann die Schnittstellenkarte 40 die minimalen und maximalen Zeiten zwischen Datenerneuerungen bestimmen, und zwar für bestimmte publizierte Daten, und kann bestimmen, ob diese Zeiten innerhalb eines zulässigen Bereiches liegen, um festzustellen, ob Kommunikationsprobleme existieren. In ähnlicher Weise kann die Schnittstellenkarte 40 die Zeiten, zu welchen bestimmte Daten angenommenenmaßen auf dem Bus zu veröffentlichen sind, mit der tatsächlichen Zeit vergleichen, zu welcher diese Daten auf dem Bus veröffentlicht werden, kann verbrauchte Datenzählwerte für die Funktionsblöcke oder Vorrichtungen überwachen und kann irgendwelche anderen ge-

wünschten Statistiken aufbewahren, um Zeitsteuerungs- oder andere Kommunikationsfehler auf dem Bus zu detektieren. Natürlich können irgendwelche anderen publizierten Daten überwacht werden, um die Kommunikations- oder Zeitsteuerprobleme auf dem Bus zu bestimmen. Wie oben dargelegt wurde, können die statistischen Informationen, die durch die Schnittstellenkarte 40 erzeugt oder gespeichert werden, für irgendeinen Funktionsblock, Vorrichtung oder Segment des Busses aufbewahrt werden und können zu dem Controller (oder irgendeiner anderen Vorrichtung) für Diagnosezwecke gesendet oder von dem Controller gelesen werden.

Fig. 6 veranschaulicht ein Flußdiagramm 200, welches den Weg anzeigt, in welchem ein Schattenfunktionsblock innerhalb des Controllers 12 erstellt werden kann. Während die Flußdiagramme der Fig. 6-12 eine Anzahl von Blöcken veranschaulichen, sei darauf hingewiesen, daß diese Blöcke lediglich eine Sequenz von Schritten angeben, die auszuführen sind, und daß die Schritte in der Software oder in irgendeiner anderen gewünschten Weise ausgeführt werden können. Die Flußdiagrammblöcke der Fig. 6-12 stellen jedoch nicht notwendigerweise Funktionsblöcke oder Controllerblöcke dar, ähnliche den Funktionsblöcken, die in Fig. 4 veranschaulicht sind.

Bei einem Block 201 empfängt der Controller 12 ein Modul-Installationsskript von der Anwenderworkstation, die auf einem der PCs 14 von Fig. 1 bestehen kann. Das Modul-Installationsskript wird durch das Konfigurationswerkzeug erzeugt, welches durch die Anwenderworkstation abläuft oder auf dieser Anwenderworkstation läuft, und enthält alle Informationen, die erforderlich sind, um die Objekte (in einer objektorientierten Programmiersprache) zu erstellen, die den Funktionsblöcken eines Steuermoduls in dem Controller 12 zugeordnet sind. Das heißt, das Installationsskript konfiguriert die Blöcke (wie beispielsweise die Funktionsblöcke, die dem Steuermodul 100 von Fig. 4 zugeordnet sind) und die Verbindungen zwischen solchen Blöcken, die durch die Verbindungen (Links) definiert sind. Natürlich werden all diese Informationen durch den Anwender geliefert, während dieser das Konfigurationswerkzeug verwendet. Wenn ein Funktionsblock durch einen externen Funktionsblock zu implementieren ist, wie beispielsweise einen in einer Bereichsvorrichtung, erzeugt ein Block 202 einen Schattenfunktionsblock innerhalb des Controller 12, und zwar durch Erzeugen eines Objektes (innerhalb einer objektorientierten Programmiersprache), welches ähnlich ist den Funktionsblöcken für andere Vorrichtungen, die innerhalb des Controllers gelegen sind, ausgenommen dieses führt keinerlei Steuerfunktionen durch. Während die Schattenfunktionsblöcke in bevorzugter Weise als ein Objekt in einer objektorientierten Programmierungsumgebung erzeugt werden, können sie unter Verwendung irgendeiner anderen Programmierungsumgebung erzeugt werden, die dem Controller 12 zugeordnet ist oder von diesem verwendet wird.

Wenn ein Schattenfunktionsblock aufgebaut wird, veranlaßt der Block 202 die Schnittstellenkarte 40, den tatsächlichen Funktionsblock innerhalb der Fieldbus-Vorrichtung abzutasten und Informationen von dem tatsächlichen Funktionsblock zu erhalten, die erforderlich sind, um zu Beginn den Schattenfunktionsblock zu konfigurieren. Ferner erstellt der Block 202 die Datenbankstellen innerhalb der unteren Datenbank 156 des Controllers 12, an die die Schnittstellenkarte 12 Betrachtungsobjekt- und Warnobjektdateien als auch verkettete Parameterdaten eingeben sollte, die von dem tatsächlichen Funktionsblock erhalten werden. Der Block 202 informiert auch die Schnittstelle 40 darüber, wie oft Betrachtungsobjekt- und Warnobjektinformationen basierend auf der Abtastrate des Moduls 100 angefragt werden sollen.

Nachdem der Block 202 einen Schattenfunktionsblock in dem Kontroller erzeugt hat und die geeigneten Verbindungen zwischen der Datenbank 156, dem Schattenfunktionsblock 108 und der Schnittstellenkarte 40 identifiziert hat, konfiguriert ein Block 203 die untere Datenbasis 156, um die gespeicherten verketteten Parameterdaten (das heißt diejenigen, die durch die Schnittstellenkarte 40 unter Verwendung des synchronen Veröffentlichlicher/Teilnehmer-VCRs erhalten wurden) an den Schattenfunktionsblock zu liefern, anstelle der Daten, die für die verketteten Parameter unter Verwendung der Objektoperation erhalten werden. Dies ist erforderlich, um sicherzustellen, daß die zuletzt erstellten Daten für die verketteten Parameter (das heißt diejenigen, die durch die synchronen Kommunikationen in dem Feldbus-Netzwerk erhalten wurden) dem Schattenfunktionsblock 108 angeboten werden anstelle der Daten, die für diese Parameter unter Verwendung der Betrachtungslistenoperation erhalten werden (was asynchron erfolgt).

Wenn ein Steuermodul, der einen Schattenfunktionsblock verwendet, zu Beginn konfiguriert wird, muß das Feldbus-Kommunikationsnetzwerk ebenfalls konfiguriert werden, um die erforderlichen synchronen und asynchronen Nachrichtenübertragungen zwischen dem aktuellen Funktionsblock 112 und dem Schattenfunktionsblock 108 zu unterstützen. Die Fig. 7-12 veranschaulichen Flußdiagramme, welche die Schritte herausgreifen, die beim Konfigurieren des Feldbus-Netzwerks involviert sind, um die Schattenfunktionsblöcke zu unterstützen. Während die Fig. 7-10 als getrennte Flußdiagramme dargestellt sind, können sie gleichzeitig oder nahezu gleichzeitig ausgeführt werden.

Fig. 7 zeigt ein Flußdiagramm 210, welches dazu verwendet werden kann, einen verbindungsaktiven Plan oder Ablaufsteuerung in dem Feldbus-Netzwerk zu erstellen. Ein Block 211 innerhalb des Kontrollers 12 empfängt das LAS-Plan-Installationsskript, wie es durch das Konfigurationswerkzeug für das Feldbus-Kommunikationsnetzwerk entwickelt wurde, und zwar nach Inbetrachtziehung aller synchroner Nachrichtenübertragungen, die über den Feldbus-Bus stattfinden müssen, inklusive solcher, die für die Schattenfunktionsblöcke erforderlich sind. Natürlich kann dieses Installationsskript unter Verwendung bekannter Werkzeuge erzeugt werden, die für das Feldbus-Kommunikationsprotokoll verfügbar sind, die auch in dem Konfigurationswerkzeug des Kontrollers 12 integriert sein können. Ein Block 212 installiert dann den LAS-Plan (Abwickler) an dem angepeilten Feldbus-Port, der beispielsweise die Schnittstellenkarte 40 aus Fig. 5 sein kann.

Danach erzeugt ein Block 213 automatisch Teilnehmerverbindungen und VCRs innerhalb des Feldbus-Netzwerks basierend auf den Daten, die durch die Feldbus-Vorrichtungen veröffentlicht werden. Speziell werden diese Teilnehmerverbindungen derart erzeugt, daß die Schnittstellenkarte 40 teil hat an den verketteten Parametern der Feldbus-Funktionsblöcke, für Schattenfunktionsblöcke innerhalb des Kontrollers 12 existieren. In ähnlicher Weise gibt die Schnittstellenkarte 40 Daten heraus, die durch die Funktionsblöcke innerhalb des Kontrollers 12 erzeugt werden und die zu den Schattenfunktionsblöcken innerhalb des Kontrollers 12 als ein verketteter Parameter gesendet werden. Allgemein gesagt, agiert die Schnittstellenkarte 40 als ein Funktionsblock-Proxy für die verketteten Daten zu und von den Schattenfunktionsblöcken. Ferner wirkt die Schnittstellenkarte 40 als ein Funktionsblock-Proxy, um nach Betrachtungsobjekt- und Warnobjektinformationen von den tatsächlichen oder aktuellen Funktionsblöcken anzufragen, für welche Schattenfunktionsblöcke existieren, um die nicht verketteten Daten zu erneuern, die den Schattenfunktionsblöcken innerhalb des Kontrollers 12 zugeordnet sind.

Ein Block 214 bildet dann den LAS-Installationsstatus in dem Kontroller (oder DeltaV-)Status ab, so daß der Kontroller 12 bestimmen kann, ob die LAS-Installation akzeptiert worden ist oder durch das Feldbus-Netzwerk abgewiesen wurde. Dies ermöglicht es dem Kontroller 12 zu verifizieren, ob die Installation des Feldbus-Netzwerks stattgefunden hat.

Fig. 8 veranschaulicht ein Flußdiagramm 220, welches die Schritte zeigt, die dazu verwendet werden, um die Veröffentlichlicher-Verbindungen innerhalb des Feldbus-Netzwerks zu erstellen. Ein Block 221 in dem Kontroller 12 empfängt die Veröffentlichlicher-Verbindungen, die durch das Workstation-Konfigurationswerkzeug für das Feldbus-Netzwerk erzeugt wurden, und ein Block 222 sendet dann diese Veröffentlichlicher-Verbindungen (publisher links) zu der Schnittstellenkarte 40, um die Veröffentlichlicher-Verbindungen zu erstellen und auch die zugeordneten VCRs, die erforderlich sind, um die Schattenfunktionsblockkommunikationen als auch andere Kommunikationen auf dem Funktionsblock 42 zu unterstützen.

Fig. 9 zeigt ein Flußdiagramm 230, welches die Schritte veranschaulicht, die der Installation einer Vorrichtungskonfiguration innerhalb einer Feldbus-Vorrichtung zugeordnet sind. Ein Block 231 empfängt das Vorrichtungskonfigurations-Installationsskript von der Workstation, konfiguriert durch das Konfigurationswerkzeug, nachdem all die Verbindungs- und anderen Informationen für diese Vorrichtung erstellt worden sind. Ein Block 232 löscht dann die frühere Konfiguration in der Vorrichtung, indem sie geeignete Befehle zu der Vorrichtung über die Schnittstellenkarte 40 sendet. Während es nicht strikt erforderlich ist, wurde es als wünschenswert gefunden, die frühere Vorrichtungskonfiguration zu löschen, bevor versucht wird, eine neue Vorrichtungskonfiguration in der Feldbus-Vorrichtung zu installieren, um dadurch Installationsprobleme zu vermeiden.

Ein Block 233 installiert dann die VCRs und ein Block 234 installiert die Verbindungen, die erforderlich sind, um die Kommunikation gemäß dem verbindungsaktiven Plan und die Veröffentlichlicher/Teilnehmerbeziehungen, die für das Feldbus-Netzwerk erstellt wurden, zu implementieren. Ein Block 235 installiert dann die Feldbus-Startliste in der Vorrichtung, was die Funktionsblock-Programmausführung von dieser Vorrichtung mit der Programmausführungen von anderen Funktionsblöcken in anderen Vorrichtungen des Feldbus-Netzwerks synchronisiert. Danach installiert ein Block 236 den Makrozyklus der Vorrichtung, nachdem dieser Makrozyklus berechnet worden ist oder bestimmte worden ist, um all den synchronen Kommunikationen Rechnung zu tragen, die zwangsweise über den Bus 42 stattfinden müssen, welcher dem Feldbus zugeordnet ist.

Fig. 10 zeigt ein Flußdiagramm 240, welches die Schritte veranschaulicht, die dazu verwendet werden, um einen speziellen oder bestimmten Funktionsblock innerhalb einer bestimmten Vorrichtung innerhalb des Feldbus-Netzwerks zu erstellen. Ein Block 241 empfängt zuerst ein Funktionsblock-Installationsskript, wie es durch das Konfigurationswerkzeug entwickelt wurde. Ein Block 242 installiert dann den nächsten Funktionsblock und Programmausführungsperiodenparameter, wie dies in typischer Weise vorgenommen wird, wenn ein Feldbus-Funktionsblock konfiguriert wird. Danach setzt ein Block 243 den Funktionsblockmodus außer Service, was erforderlich ist, um die Werte innerhalb des Funktionsblockes zu ändern. Ein Block 244 installiert dann die gemäß dem Funktionsblockparameter konfigurierten Werte oder die Anwenderprioritäten (overrides) (das heißt die anwenderdefinierten Werte) innerhalb des Funktionsblockes. Diese durch den Anwender konfigurierten Werte können in irgendeiner Weise erstellt werden, wie beispiels-

weise durch die Verwendung von Dialogboxen in der Workstation während der Konfiguration des Prozeßregelsystems. Danach stellt ein Block 245 den Funktionsblockmodus auf den konfigurierten Wert ein, so daß der Funktionsblock in Einklang mit der Art arbeitet, in welcher dieser konfiguriert ist.

Es sei darauf hingewiesen, daß die Flußdiagramme der Fig. 7-9 lediglich die normalen Schritte angeben, die dazu verwendet werden, um ein Fieldbus-Netzwerk zu konfigurieren. In diesem Fall trägt jedoch die Erstellung des Fieldbus-Netzwerks Rechnung hinsichtlich der und enthält die Veröffentlichlicher/Teilnehmerverbindungen und VCRs, die erforderlich sind, um den Betrieb von irgendeinem der Schattenfunktionsblöcke innerhalb des Controllers 12 zu unterstützen. Natürlich kann die Anwenderworkstation oder der Controller 12 das Fieldbus-Netzwerk automatisch konfigurieren, basierend auf den Informationen, die durch den Anwender über die Erstellung des Regelsystems während der Konfiguration dieses Systems geliefert werden.

Gemäß Fig. 11 veranschaulicht ein Flußdiagramm 250 die Betriebsweise eines Schattenfunktionsblockes, wenn ein Steuermodul, der diesen Schattenfunktionsblock enthält, auf dem Controller 12 läuft, um die Prozeßsteuerung oder -regelung durchzuführen. Wie zu erkennen ist, implementiert der Controller 12 die Blöcke innerhalb eines bestimmten Moduls (wie beispielsweise die Blöcke 102, 104, 106 und 108 von Fig. 4) in der Modulabtastrate, die dem Modul zugeordnet ist. Die Kommunikation von verketteten Daten (wie denjenigen, die durch die Verbindungen in dem Diagramm von Fig. 4 definiert sind) zwischen den aktuellen Fieldbus-Funktionsblöcken und den Schattenfunktionsblöcken erfolgt in der synchronen Makrozyklusrate des Fieldbus-Netzwerks, die verschieden sein kann von der Modulabtastrate des Controllers 12. Als ein Ergebnis werden die Daten für die verketteten Parameter in typischer Weise in der unteren Datenbank 156 des Controllers 12 in einer unterschiedlichen (gewöhnlich schnelleren) Rate als die Daten für nicht verkettete Parameter gespeichert.

Wenn der Schattenfunktionsblock in dem Controller 12 implementiert ist, tastet ein Block 252 die Betriebsdaten ab, die innerhalb der unteren Datenbank 156 des Controllers 12 gespeichert sind. Das heißt, es werden während jeder Modulperiode die Betriebsdaten, die in der unteren Datenbank 156 gespeichert wurden, durch den Schnittstellenmodul 40 gelesen. Ein Block 254 ermittelt, ob die Abtastung voranschreitet. Wenn dies nicht der Fall ist, erneuert ein Block 256 den Blockfehler in dem Schattenfunktionsblock, stellt den Schattenfunktionsblock-Ausgangsstatus auf schlecht ein und stellt die Port-Integrität so ein, um dem Controller 12 anzuzeigen, daß der Schattenfunktionsblock ausgefallen ist und daß daher ein Problem in bezug auf den externen Funktionsblock oder in bezug auf Nachrichtenübertragungen mit dem externen Funktionsblock innerhalb des Fieldbus-Netzwerks auftreten kann. Wenn solch ein Fehler detektiert wird, werden die Schattenfunktionsblockdaten nicht erneuert (da diese Daten alt sind oder in jedem Fall schlechte Daten sind) und es wird die Erneuerungsoperation der Daten innerhalb des Schattenfunktionsblocks für diesen Modulabastzyklus angehalten. Wenn jedoch die Abtastung voranschreitet, kopiert ein Block 258 die empfangenen Betriebsdaten in dem Schattenfunktionsblock. Natürlich enthalten die Betriebsdaten nicht nur die Daten, die durch die Betrachtungs- und Warnobjekte in der Modulabtastrate erhalten werden, sondern auch die Daten für die verketteten Parameter, die erhalten wurden und in die untere Datenbank 156 in einer Rate eingeschrieben werden, die durch den Makrozyklus des Fieldbus-Netzwerks festgelegt wird.

Ein Block 260 ermittelt dann, ob der statistische Revisi-

onsparameter, der dem tatsächlichen Funktionsblock zugeordnet ist (der durch die Betrachtungsobjektoperation geliefert wird) zugenommen hat. Wenn dies der Fall ist, instruiert ein Block 252 die Schnittstellenkarte 40, die statischen Daten, die dem Fieldbus-Funktionsblock zugeordnet sind, zu lesen und diese statischen Daten zu der unteren Datenbank 156 zu liefern, wo diese Daten gelesen werden und in den Schattenfunktionsblock eingesetzt werden. Wie dies bekannt ist, zeigt der statische Visionsparameter an, ob irgendwelche statischen Daten, die normalerweise durch die statische Betrachtungsliste vorgesehen werden, geändert wurden. Wenn der statische Revisionsparameter nicht erhöht wurde, werden keine der statischen Daten geändert und es ist nicht erforderlich, diese Daten während jedes Zyklus des Steuermoduls zu lesen. Wenn jedoch die statische Datenrevision zugenommen hat, dann haben sich einige der statischen Daten geändert und es müssen diese statischen Daten gelesen werden und in den Schattenfunktionsblock gesetzt werden, so daß der Schattenfunktionsblock die Daten spiegelt, die tatsächlich innerhalb des externen Funktionsblocks vorhanden sind.

Nachdem die statischen Daten erhalten worden sind oder die statische Revision nicht zugenommen hat, bildet ein Block 264 die Fieldbus-Alarme (und andere Daten, wenn dies erforderlich ist) in den Alarmen (oder anderen Datenbereichen) des Controllers 12 ab. Diese Abbildung kann unter Verwendung einer Nachschlagetabelle erreicht werden oder durch irgendeine andere Abbildungstechnik. Die abgebildeten Alarmgrößen (und andere Daten) werden dann in dem Schattenfunktionsblock gespeichert, um durch andere Steuerblöcke des Steuermoduls verwendet zu werden. Die Alarmgrößen können auch für einen Anwender dargestellt werden oder in irgendeiner anderen gewünschten Weise verwendet werden.

Danach ermittelt ein Block 266, ob die Daten für die verketteten Parameter veraltet sind (das heißt nicht mehr aktuell sind). Solch eine Anzeige wird in regulärer Weise in den Fieldbus-Kommunikationen erzeugt und sie zeigt an, daß die empfangenen Daten nicht in dem allerletzten Makrozyklus des Fieldbus-Netzwerks erzeugt wurden, was anzeigen kann, daß ein Zeitsteuerproblem oder ein anderes Problem innerhalb des Fieldbus-Netzwerks existieren kann. Wenn die Verkettungsparameterdaten veraltet sind, markiert ein Block 268 die Ausgangsdaten als BadNotConnected, was den Modus oder Status von anderen Funktionsblöcken innerhalb des Controllers 12 ändern kann. Danach oder wenn die verketteten Daten nicht veraltet sind, macht ein Block 270 die Betriebsdaten für andere Blöcke innerhalb des Controllers 12 sichtbar und der Controller 12 fährt damit fort, basierend auf den neuen Betriebsdaten zu arbeiten.

Wie ersehen werden kann, veranschaulicht das Flußdiagramm von Fig. 11 die Betriebsweise der Erneuerung des Schattenfunktionsblocks, um sicherzustellen, daß dieser die Daten spiegelt, die dem externen Funktionsblock innerhalb einer externen Vorrichtung zugeordnet sind. Jedoch kann der Schattenfunktionsblock auch Daten zu dem externen Funktionsblock übertragen und Schreibbefehle an den externen Funktionsblock senden. Solche Daten und Schreibbefehle können von einem Anwender, beispielsweise bei einer Workstation, vorgesehen werden oder können an anderen Funktionsblöcken innerhalb des Controllers 12 entspringen und von diesen ausgesendet werden, und zwar über die erstellten Verbindungen. Gemäß Fig. 12 veranschaulicht ein Flußdiagramm 280 die Schritte, die unternommen werden, um Daten oder Schreibbefehle zu einem externen Funktionsblock über den Schattenfunktionsblock zu senden. Bei einem Block 281 schreibt ein Steuerblock innerhalb des Controllers 12 Daten über eine Eingangsverbindung in den

Schattenfunktionsblock ein. Alternativ kann ein Anwender einen Schreibbefehl an den Schattenfunktionsblock über eine Anwenderschnittstelle senden, was es einem Anwender ermöglicht, z. B. von Hand einen Sollwert oder einen anderen Wert, der dem externen Funktionsblock zugeordnet ist, zu ändern. Der Schattenfunktionsblock sendet dann unmittelbar einen Schreibbefehl oder andere Daten an die Schnittstellenkarte 40. Wenn solche Daten oder Befehl einem verketteten Parameter zugeordnet sind oder ist, veröffentlicht die Schnittstellenkarte 40 die Daten auf dem Feldbus-Bus für den externen Funktionsblock zu einem geeigneten oder synchronen Zeitpunkt, wie dieser durch den Makrozyklus des Feldbus-Netzwerks festgelegt ist. Wenn der Schreibbefehl oder die Daten nicht einem verketteten Parameter zugeordnet ist bzw. sind, verwendet die Schnittstellenkarte 40 asynchrone Nachrichtenübertragungen, um den Befehl oder die Daten an den externen Funktionsblock zu übermitteln. Danach empfängt der externe Funktionsblock die Daten oder den Befehl und erneut deren oder dessen Attributparameter entsprechend. Diese Änderungen werden dann über die Schnittstellenkarte 40 unter Verwendung von Veröffentlichlicher/Teilnehmer-Kommunikationen zurück übertragen, als auch die Betrachtungsobjektoperation, und es werden die neuen Daten in die untere Datenbank 156 gesetzt, wo dann während des nächsten Abtastzyklusses des Steuermoduls diese Daten in den Schattenfunktionsblock eingesetzt werden.

Wenn der Block 282 eine Schreibanfrage an die Schnittstellenkarte 40 sendet und diese Anfrage zu dem externen Funktionsblock gesendet wird, empfängt die Schnittstellenkarte 40 eine Antwort von dem Funktionsblock, die anzeigt, ob der Schreibvorgang vervollständigt worden ist oder ob die Daten angenommen oder empfangen wurden. Die Schnittstellenkarte 40 sendet ihrerseits diese Antwort zu dem Schattenfunktionsblock. Wenn der Schreibvorgang fehlgeschlagen ist, kann die Status-, Alarm- oder Fehlereinstellung des Schattenfunktionsblocks geändert werden, um diesen Fehler zu reflektieren. Wenn beispielsweise eine Schreibanfrage, die einem verketteten Parameter zugeordnet ist, nicht empfangen wurde oder nicht richtig durch die Schnittstelle 40 implementiert wurde, kann dies in dem Fehlerstatus des Schattenfunktionsblocks angezeigt werden. Wenn es gewünscht wird, wenn ein Schreibbefehl, der von einem Anwender stammt, fehlschlägt, implementiert zu werden, kann ein Block 283 eine Anzeige über den Fehlschlag für den Anwender bei der Workstation senden oder zu irgendeiner anderen Anwenderschnittstelle, um dadurch den Anwender über den fehlgeschlagenen Versuch zu unterrichten, in den externen Funktionsblock zu schreiben.

Während sich die Beschreibung auf die Implementierung und die Verwendung eines einzelnen Schattenfunktionsblockes 108 gerichtet hat, sei darauf hingewiesen, daß viele Schattenfunktionsblöcke in dem gleichen Steuermodul oder Controller implementiert werden können, um es dem Steuermodul oder Controller zu ermöglichen, viele externe Funktionsblöcke zu verwenden. Ferner können Schattenfunktionsblöcke implementiert werden unter Verwendung irgendeines externen Prozeßsteuer- oder -regel-Kommunikationsprotokolls (neben dem Feldbus-Protokoll) und können dazu verwendet werden, mit irgendeinem Typ eines Funktionsblocks zu kommunizieren, inklusive irgendeinem Funktionsblock, der ähnlich ist mit oder der gleiche ist wie irgendeiner der unterschiedlichen Funktionsblöcke, die durch das Feldbus-Protokoll spezifisch identifiziert und unterstützt werden. Obwohl darüber hinaus Schattenfunktionsblöcke hier so beschrieben wurden, als ob sie einem externen Funktionsblock zugeordnet sind, und zwar in der Form, in welcher das Feldbus-Protokoll einen "Funktionsblock"

definiert, sei darauf hingewiesen, daß die Verwendung des Ausdrucks "Funktionsblock" hier nicht auf das beschränkt ist, was das Feldbus-Protokoll als einen Funktionsblock definiert, sondern statt dessen irgendein anderer Typ eines Blocks, Programms, einer Hardware, Firmware usw. enthalten ist, die mit irgendeinem Typ eines Steuersystems oder Regelsystems und/oder Kommunikationsprotokoll zugeordnet ist und die dazu verwendet werden kann, um eine gewisse Steuerfunktion oder Regelfunktion zu implementieren. Obwohl somit Funktionsblöcke typisch die Form von Objekten annehmen, und zwar innerhalb einer objektorientierten Programmierungsumgebung, muß dies nicht der Fall sein und statt dessen können andere logische Einheiten dazu verwendet werden, um eine bestimmte Steuerung oder Regelung (inklusive Eingabe- und Ausgabe-)Funktionen innerhalb einer Prozeßsteuerumgebung oder Prozeßregelumgebung auszuführen.

Obwohl ferner der Schattenfunktionsblock, der hier beschrieben wurde, in bevorzugter Weise in einer Software implementiert wird, die beispielsweise in einem Controller oder einer anderen Prozeßsteuervorrichtung gespeichert ist, kann sie alternativ oder zusätzlich in einer Hardware, Firmware usw. in gewünschter Weise implementiert sein. Wenn sie in einer Software implementiert ist, kann der Schattenfunktionsblock der vorliegenden Erfindung in irgendeinem computerlesbaren Speicher gespeichert sein, wie beispielsweise einer Magnetplatte, einer Laserplatte oder einem anderen Speichermedium, in einem RAM oder ROM eines Computers usw. In ähnlicher Weise kann diese Software an einen Anwender oder eine Vorrichtung aus geliefert werden, und zwar über irgendein bekanntes oder gewünschtes Auslieferungsverfahren, inklusive beispielsweise über einen Nachrichtenübertragungskanal, wie beispielsweise eine Telefonleitung, das Internet usw.

Obwohl auch der Schattenfunktionsblock der vorliegenden Erfindung in Einzelheiten in Verbindung mit einem Prozeßregelnetzwerk beschrieben wurde, welches die Prozeßsteuerfunktionen in einer dezentralisierten oder verteilten Weise unter Verwendung eines Satzes von Feldbus-Vorrichtungen implementiert, sei darauf hingewiesen, daß der Schattenfunktionsblock der vorliegenden Erfindung in Verbindung mit Prozeßregelnetzwerken verwendet werden kann, die Steuerfunktionen ausführen, unter Verwendung von anderen Typen von Bereichsvorrichtungen und Kommunikationsprotokollen, inklusive der Protokolle, die auf anderen als den zwei Leitungsbussen basieren, und Protokollen, die analog und/oder digitale Nachrichtenübertragungen unterstützen. Der Schattenfunktionsblock der vorliegenden Erfindung kann beispielsweise in irgendeinem Prozeßregelnetzwerk verwendet werden, welches Vorrichtungen verwendet, die in Einklang mit dem HART-, PROFIBUS-, usw. Kommunikationsprotokollen oder irgendeinem anderen Kommunikationsprotokoll stehen, die es nunmehr gibt oder die in der Zukunft entwickelt werden können. In ähnlicher Weise kann, wenn dies gewünscht wird, der Schattenfunktionsblock der vorliegenden Erfindung in Prozeßregelnetzwerken verwendet werden, die keine verteilten Steuerfunktionen haben, sondern statt dessen einen zentralisierten Controller oder ein Steuer- oder Regelschema verwenden, um die Vorrichtungen darin zu steuern oder zu regeln.

Während die vorliegende Erfindung in bezug auf spezifische Beispiele beschrieben wurde, die dazu dienen, die Erfindung lediglich zu veranschaulichen, jedoch nicht einzuschränken; ist es für Fachleute offensichtlich, daß Änderungen, Ergänzungen oder Weglassungen bei den offenbarten Ausführungsformen vorgenommen werden können, ohne dadurch die Idee und den Rahmen der vorliegenden Erfindung zu verlassen.

1. Interface-Funktionsblock für die Verwendung in einem Prozeßkontroller, der kommunikativ mit einer externen Vorrichtung über ein Kommunikationsnetzwerk gekoppelt ist, wobei der Prozeßkontroller folgendes implementiert:
 - eine Steueroutine unter Verwendung eines internen Funktionsblockes, welcher innerhalb des Prozeßkontrollers angeordnet ist, und
 - einen externen Funktionsblock, der in der externen Vorrichtung angeordnet ist, wobei der Interface-Funktionsblock folgendes aufweist:
 - einen Kommunikationsport mit einem Eingang, der mit dem externen Funktionsblock über das Kommunikationsnetzwerk kommuniziert, um Daten, die den externen Funktionsblock betreffen, zu empfangen, und
 - einen Speicher, der die empfangenen Daten, die den externen Funktionsblock betreffen, gemäß einem Kommunikationsprotokoll des internen Funktionsblockes speichert.
2. Interface-Funktionsblock nach Anspruch 1, der ferner einen Ausgang umfaßt, der die Daten, die den externen Funktionsblock betreffen, an den internen Funktionsblock gemäß dem Kommunikationsprotokoll des internen Funktionsblockes liefert.
3. Interface-Funktionsblock nach Anspruch 2, bei dem der Eingang des Kommunikationsports die Daten, die den externen Funktionsblock betreffen, unabhängig von der Operation des internen Funktionsblockes empfängt.
4. Interface-Funktionsblock nach Anspruch 2, bei dem der externe Funktionsblock über das Kommunikationsnetzwerk unter Verwendung eines ersten Kommunikationsprotokolls kommuniziert, welches verschieden ist von einem zweiten Kommunikationsprotokoll, das dem Konfigurationsprotokoll des internen Funktionsblockes zugeordnet ist und bei dem der Eingang des Kommunikationsports mit dem externen Funktionsblock unter Verwendung des ersten Kommunikationsprotokolls kommuniziert und der Ausgang mit dem internen Funktionsblock gemäß dem zweiten Kommunikationsprotokoll kommuniziert.
5. Interface-Funktionsblock nach Anspruch 4, bei dem das erste Kommunikationsprotokoll ein Fieldbus-Kommunikationsprotokoll ist und bei dem der Eingang des Kommunikationsports mit dem externen Funktionsblock unter Verwendung des Fieldbus-Kommunikationsprotokolls kommuniziert.
6. Interface-Funktionsblock nach Anspruch 5, bei dem der Eingang des Kommunikationsports eine Schnittstellenvorrichtung verwendet, die konfiguriert ist, um mit der externen Vorrichtung unter Verwendung des Fieldbus-Kommunikationsprotokolls zu kommunizieren, um mit dem externen Funktionsblock zu kommunizieren.
7. Interface-Funktionsblock nach Anspruch 1, bei dem der Kommunikationsport mit dem externen Funktionsblock unter Verwendung synchroner Nachrichtenübertragungen kommuniziert.
8. Interface-Funktionsblock nach Anspruch 1, bei dem der Kommunikationsport mit dem externen Funktionsblock unter Verwendung synchroner und asynchroner Nachrichtenübertragungen kommuniziert.
9. Interface-Funktionsblock nach Anspruch 1, bei dem die externe Vorrichtung unter Verwendung eines Vorrichtungskommunikationsprotokolls Nachrichten über-

mittelt, welches die Kommunikation von logisch verketteten Paketen von Daten unter Verwendung von standardisierten Kommunikationsanrufen unterstützt und bei dem der Kommunikationsport mit dem externen Funktionsblock unter Verwendung der standardisierten Kommunikationsanrufe kommuniziert, die dem Vorrichtungskommunikationsprotokoll zugeordnet sind.

10. Interface-Funktionsblock nach Anspruch 9, bei dem das Vorrichtungskommunikationsprotokoll ein Fieldbus-Kommunikationsprotokoll ist und bei dem der Kommunikationsport mit der externen Vorrichtung unter Verwendung von Betrachtungsobjekten innerhalb des Fieldbus-Kommunikationsprotokolls kommuniziert.

11. Interface-Funktionsblock nach Anspruch 1, bei dem der Kommunikationsport eine Alarmanzeige von dem externen Funktionsblock empfängt und bei dem der Speicher die empfangene Alarmanzeige speichert.

12. Interface-Funktionsblock nach Anspruch 1, bei dem der Kommunikationsport ferner einen Ausgang enthält, welcher Daten zu dem externen Funktionsblock unter Verwendung eines Kommunikationsprotokolls sendet, welches dem externen Funktionsblock zugeordnet ist.

13. Kontroller, der dafür ausgebildet ist, um kommunikativ an eine Vielzahl von Bereichsvorrichtungen gekoppelt zu werden, wobei eine der Bereichsvorrichtungen einen externen Funktionsblock enthält, der unter Verwendung eines Vorrichtungskommunikationsprotokolls Nachricht überträgt, wobei der Kontroller folgendes aufweist:

einen Prozessor;
einen Speicher; und
eine Steueroutine, die in dem Speicher gespeichert ist und durch den Prozessor implementiert wird, um die Vielzahl der Bereichsvorrichtungen zu steuern, wobei die Steueroutine folgendes enthält:

- eine Vielzahl von miteinander verbundenen internen Funktionsblöcken, die so konfiguriert sind, um ein Kontrollerprotokoll zu verwenden und durch den Kontroller implementiert sind, und
- einen Interface-Funktionsblock, der mit einer der Vielzahl der miteinander verbundenen internen Funktionsblöcke unter Verwendung des Kontrollerprotokolls kommuniziert, und der mit dem externen Funktionsblock innerhalb einer der Bereichsvorrichtungen unter Verwendung des Vorrichtungskommunikationsprotokolls kommuniziert, wobei der Interface-Funktionsblock Daten speichert, die den externen Funktionsblock betreffen, welche von dem externen Funktionsblock empfangen wurden.

14. Kontroller nach Anspruch 13, bei dem der Interface-Funktionsblock die Daten, die den externen Funktionsblock betreffen, unabhängig von der Operation der Vielzahl der internen Funktionsblöcke empfängt.

15. Kontroller nach Anspruch 13, bei dem das Vorrichtungskommunikationsprotokoll ein Fieldbus-Kommunikationsprotokoll ist und bei dem der Interface-Funktionsblock mit dem externen Funktionsblock unter Verwendung des Fieldbus-Kommunikationsprotokolls kommuniziert.

16. Kontroller nach Anspruch 15, bei dem der Interface-Funktionsblock mit dem externen Funktionsblock unter Verwendung von Betrachtungsobjekten innerhalb des Fieldbus-Kommunikationsprotokolls kommuniziert.

17. Kontroller nach Anspruch 15, bei dem der Interface-Funktionsblock mit dem externen Funktionsblock unter Verwendung von Teilnehmer /Herausgeber-Nachrichtenübertragungen innerhalb des Fieldbus-Kommunikationsprotokolls kommuniziert. 5
18. Kontroller nach Anspruch 13, bei dem der Interface-Funktionsblock Daten, welche die Eingangsgrößen und Ausgangsgrößen des externen Funktionsblockes betreffen, empfängt und speichert.
19. Kontroller nach Anspruch 13, bei dem der Interface-Funktionsblock Daten, die Alarmgrößen betreffen, welche durch den externen Funktionsblock erzeugt wurden, empfängt und speichert. 10
20. Kontroller nach Anspruch 13, bei dem der Interface-Funktionsblock Daten, die den Modus des externen Funktionsblockes betreffen, empfängt und speichert. 15
21. Kontroller nach Anspruch 13, bei dem der Interface-Funktionsblock Daten, die durch eine der Vielzahl der internen Funktionsblöcke erzeugt wurden, zu dem externen Funktionsblock sendet. 20
22. Kontroller nach Anspruch 13, bei dem der Interface-Funktionsblock einen Befehl, der durch den Kontroller erzeugt wurde, zu dem externen Funktionsblock sendet. 25
23. Verfahren zum Implementieren einer Steuer- oder Regelroutine in einem Prozeßkontroller, der kommunikativ mit einer Bereichsvorrichtung gekoppelt ist, wobei die Bereichsvorrichtung einen externen Funktionsblock aufweist, der in ihr angeordnet ist und unter Verwendung eines Vorrichtungskommunikationsprotokolls Nachrichten überträgt, wobei das Verfahren die folgenden Schritte umfaßt:
- Speichern einer Vielzahl von miteinander verbundenen internen Funktionsblöcken in dem Kontroller, der gemäß einem Kontrollerprotokoll konfiguriert ist, um diese als Teil der Steuer- oder Regelroutine zu implementieren; 30
 - Erzeugen eines Interface-Funktionsblockes innerhalb des Kontrollers, der gemäß dem Kontrollerprotokoll konfiguriert ist, wobei der Interface-Funktionsblock mit dem externen Funktionsblock unter Verwendung des Vorrichtungskommunikationsprotokolls kommuniziert; 35
 - Erzeugen einer Steuer- oder Regelroutine, welche die Vielzahl der miteinander verbundenen internen Funktionsblöcke und den Interface-Funktionsblock verwendet, um den Prozeß zu steuern oder zu regeln; und 40
 - Speichern von Daten in dem Interface-Funktionsblock während der Implementierung der Steuer- oder Regelroutine, die dem externen Funktionsblock zugeordnet sind und von diesem empfangen werden. 45
24. Verfahren nach Anspruch 23, welches ferner den Schritt der Verwendung des Interface-Funktionsblockes für eine Kommunikation mit dem externen Funktionsblock aufweist, um Daten, die dem externen Funktionsblock zugeordnet sind, unabhängig von der Operation der internen Funktionsblöcke zu erneuern. 50
25. Verfahren nach Anspruch 23, welches ferner den Schritt einer Verwendung des Interface-Funktionsblockes umfaßt, um weitere Daten zu dem externen Funktionsblock zu übertragen, um die Konfiguration des externen Funktionsblockes zu ändern. 55
26. Verfahren nach Anspruch 23, welches ferner folgende Schritte enthält:
- einem Anwender erlauben, die Steuer- oder Re-

- gelroutine dadurch zu konfigurieren, indem jeder eine Anzahl von Funktionsblöcken, die in der Steuer- oder Regelroutine zu verwenden sind, spezifiziert wird,
 - die Verbindungen zwischen den spezifizierten Funktionsblöcken, die in der Steuer- oder Regelroutine zu verwenden sind, zu identifizieren, und
 - die Lage eines bestimmten spezifizierten Funktionsblockes als einen internen Funktionsblock zu spezifizieren, der in dem Kontroller implementiert ist, oder als externen Funktionsblock zu spezifizieren, der durch die Bereichsvorrichtung implementiert ist.
27. Verfahren nach Anspruch 26, bei dem der Schritt der Spezifizierung der Lage des bestimmten spezifizierten Funktionsblockes als den externen Funktionsblock den folgenden Schritt aufweist:
- Auswählen einer externen Vorrichtung, in welcher der externe Funktionsblock zu implementieren ist, aus einer Liste von externen Vorrichtungen, die an den Kontroller angeschlossen sind.
28. Verfahren nach Anspruch 23, bei dem der Schritt der Speicherung der Daten, die dem externen Funktionsblock zugeordnet sind, den folgenden Schritt aufweist:
- Speichern einer Alarmanzeige, die durch den externen Funktionsblock in dem Interface-Funktionsblock erzeugt wird, und bei dem das Verfahren des weiteren folgenden Schritt aufweist:
 - Verwendung der Alarmanzeige, die in dem Interface-Funktionsblock gespeichert ist, zum Triggern einer Alarmverarbeitung innerhalb des Kontrollers.
29. Verfahren nach Anspruch 23, welches ferner den folgenden Schritt aufweist:
- Darstellung der Steuer- oder Regelroutine durch Darstellen einer Wiedergabe der internen Funktionsblöcke, einer Wiedergabe des Interface-Funktionsblockes und von Wiedergaben der Verbindungen zwischen den internen Funktionsblöcken und den Interface-Funktionsblöcken, so daß der Interface-Funktionsblock den externen Funktionsblock repräsentiert.

Hierzu 9 Seite(n) Zeichnungen

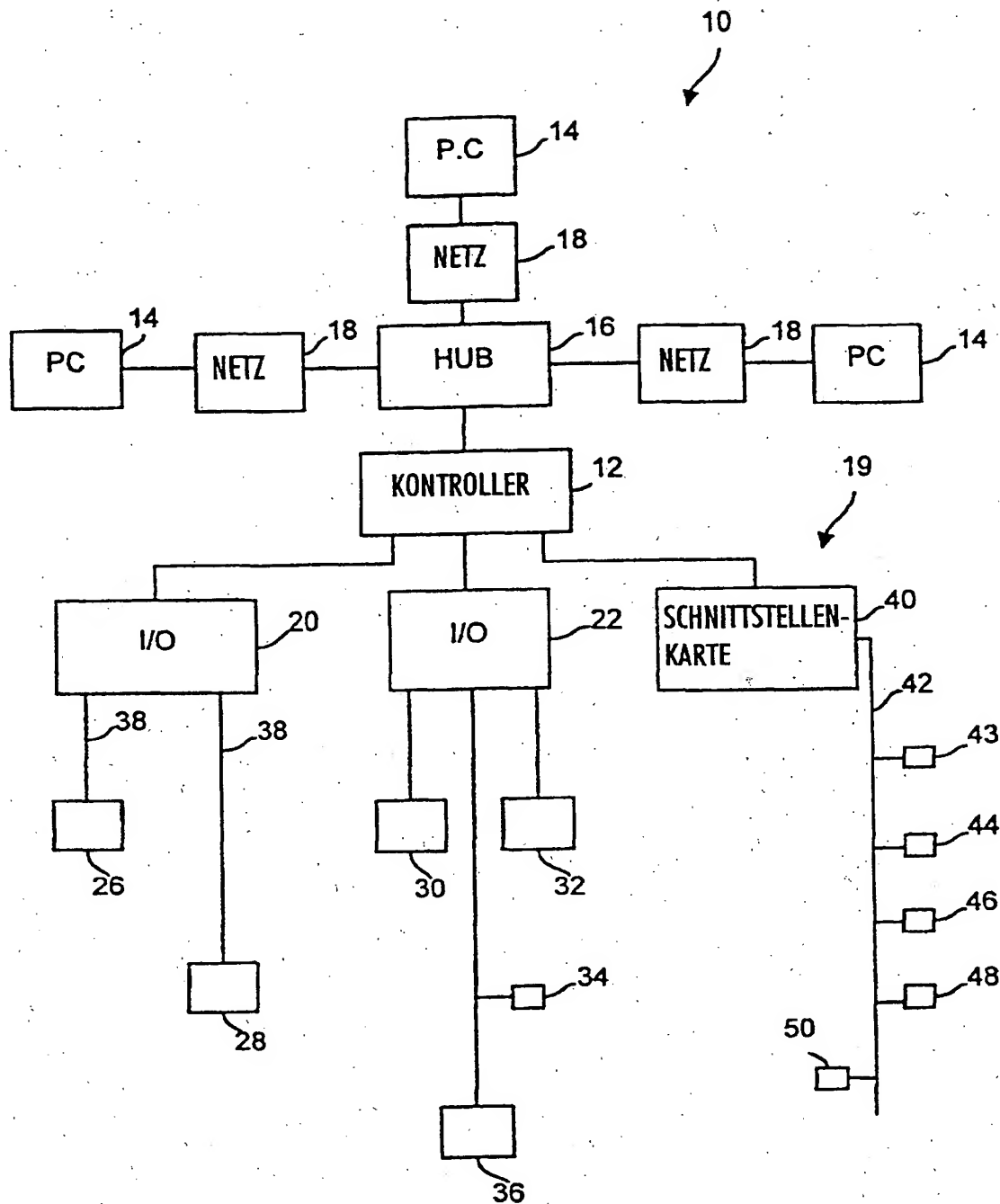


FIG. 1

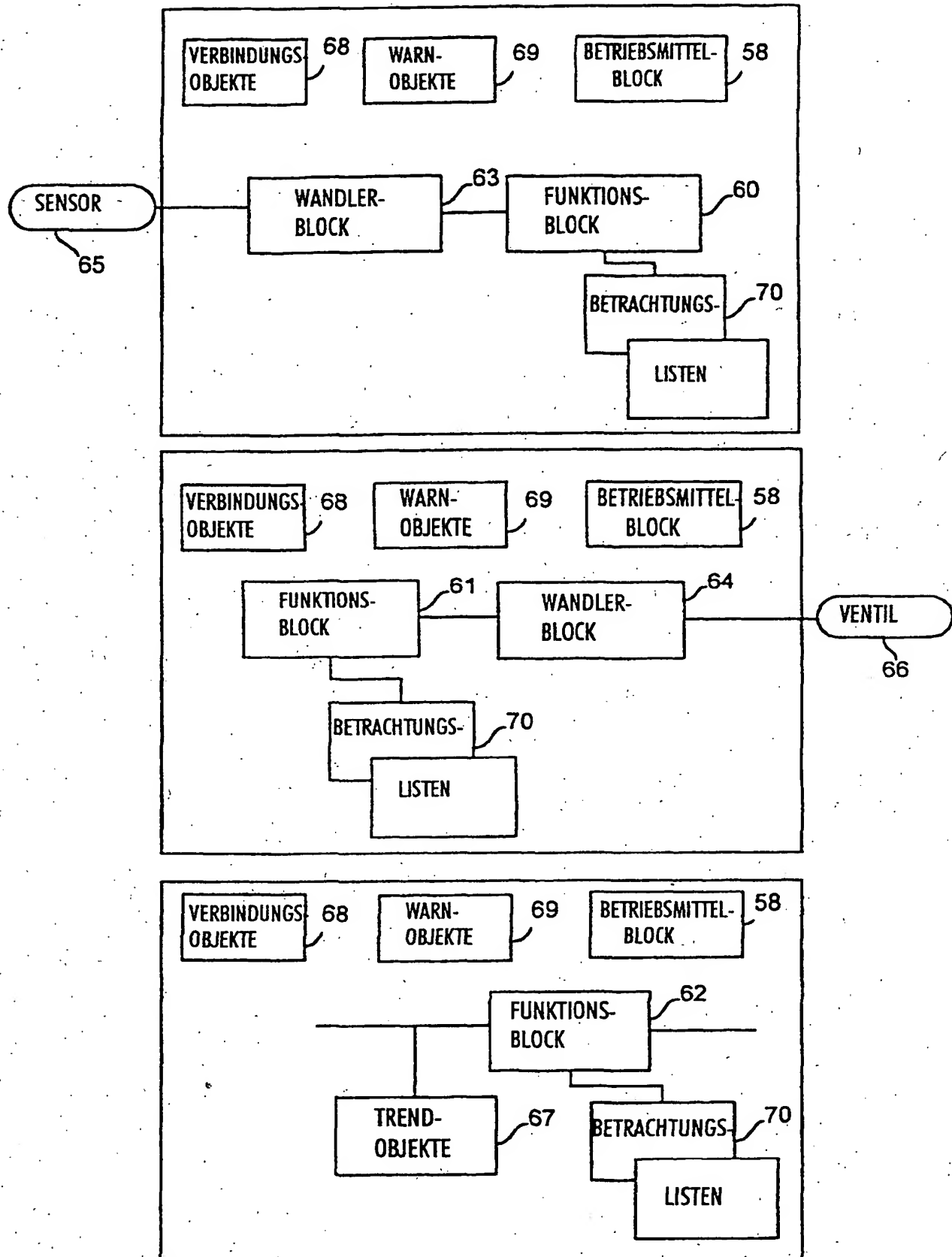


FIG. 2

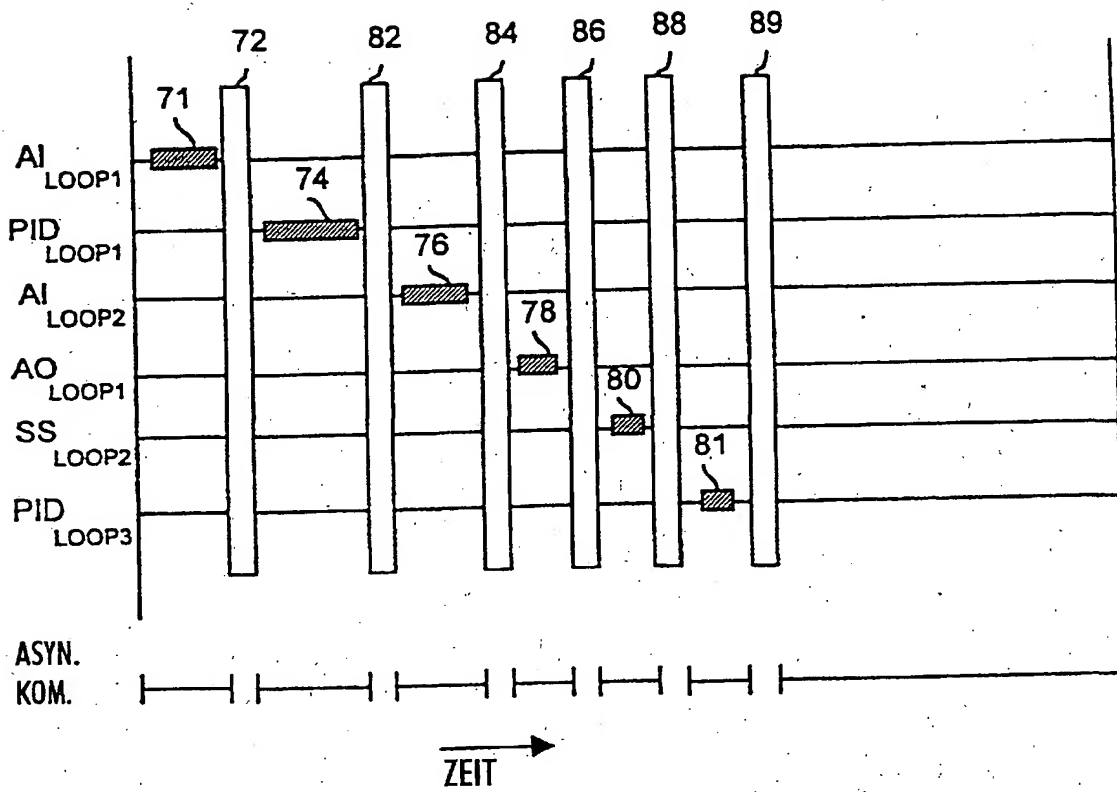


FIG. 3

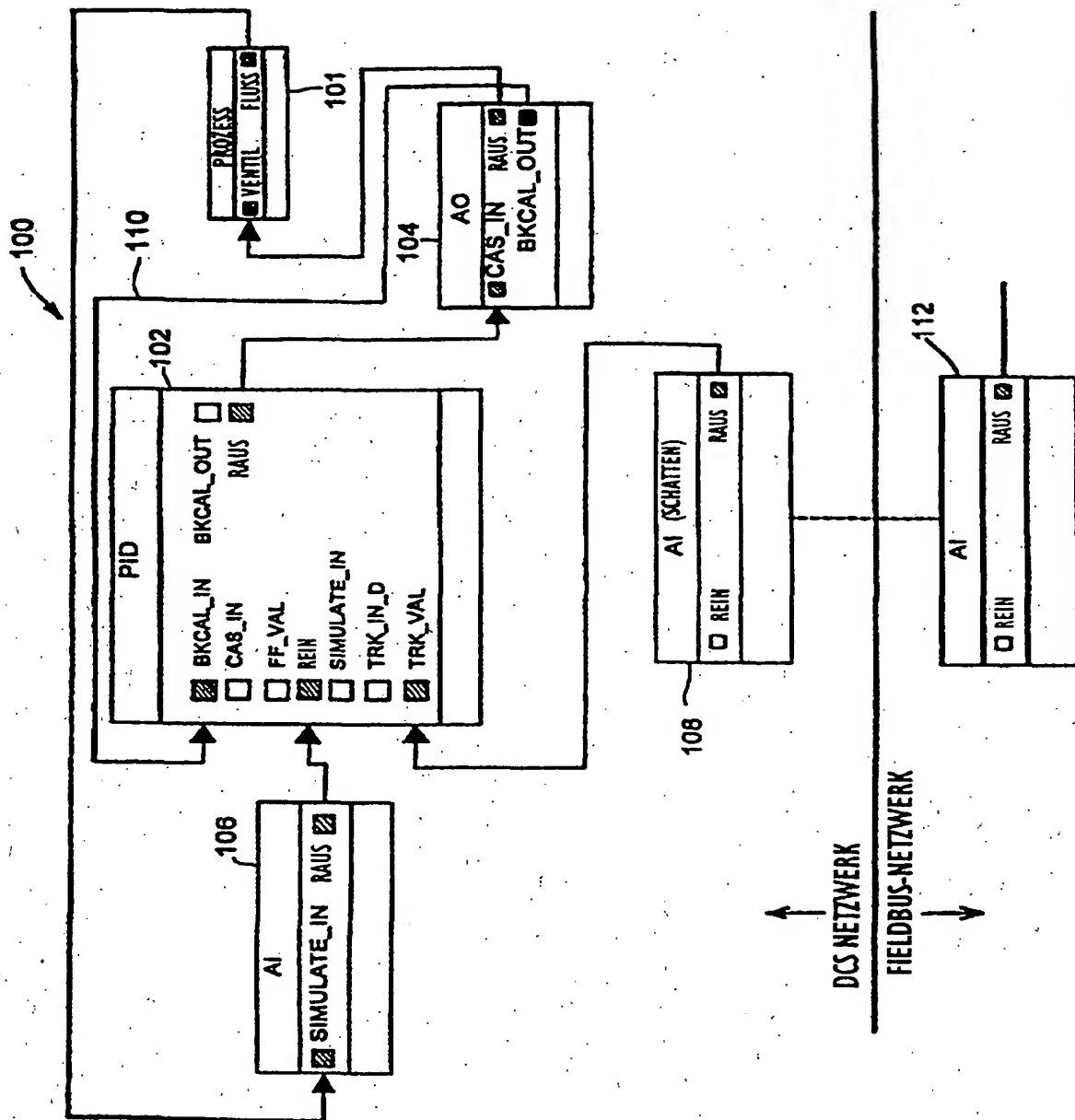


FIG. 4

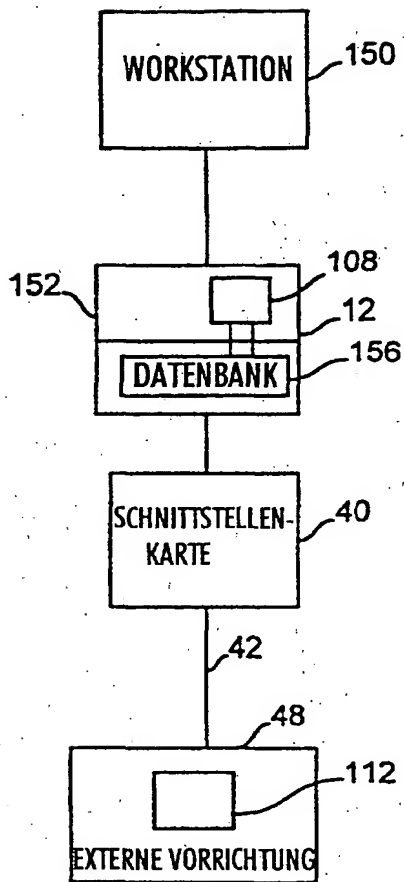


FIG. 5

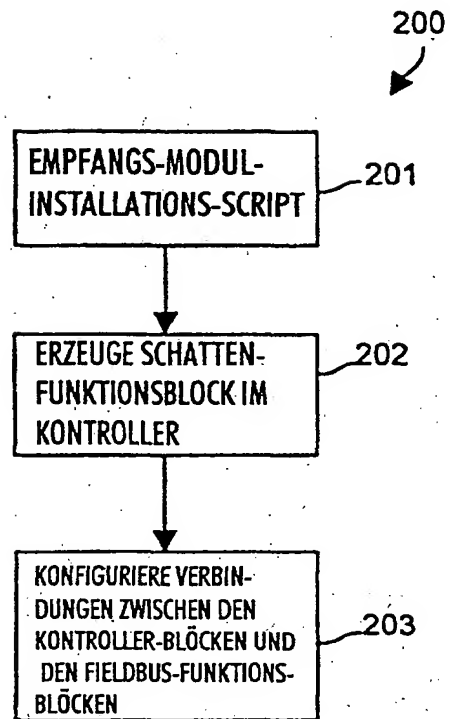


FIG. 6

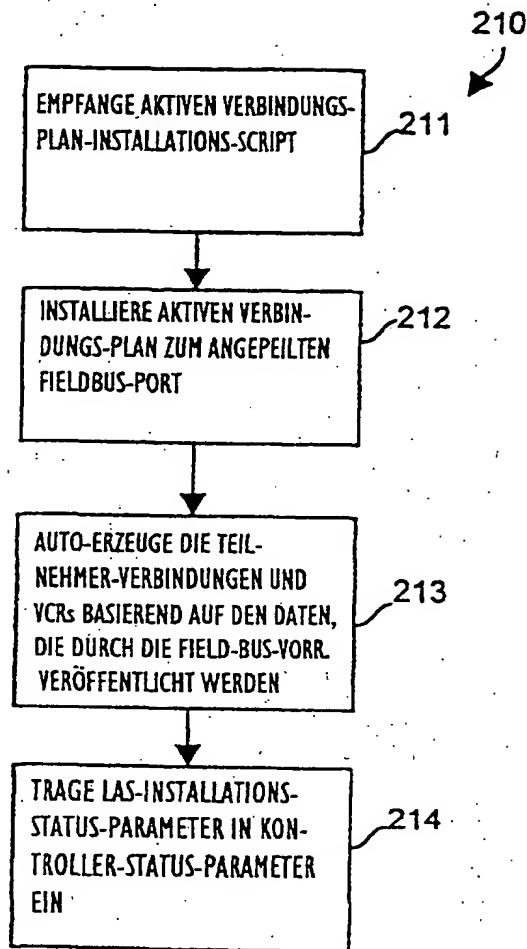


FIG. 7

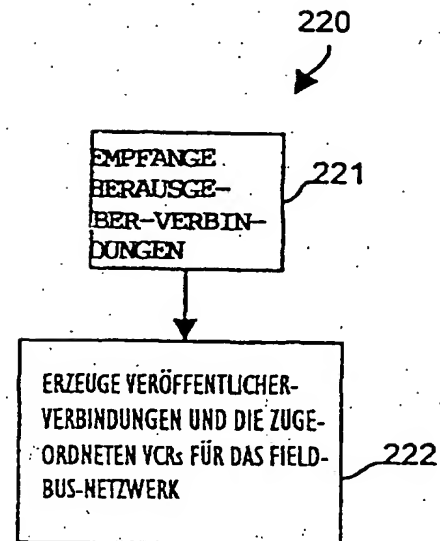


FIG. 8

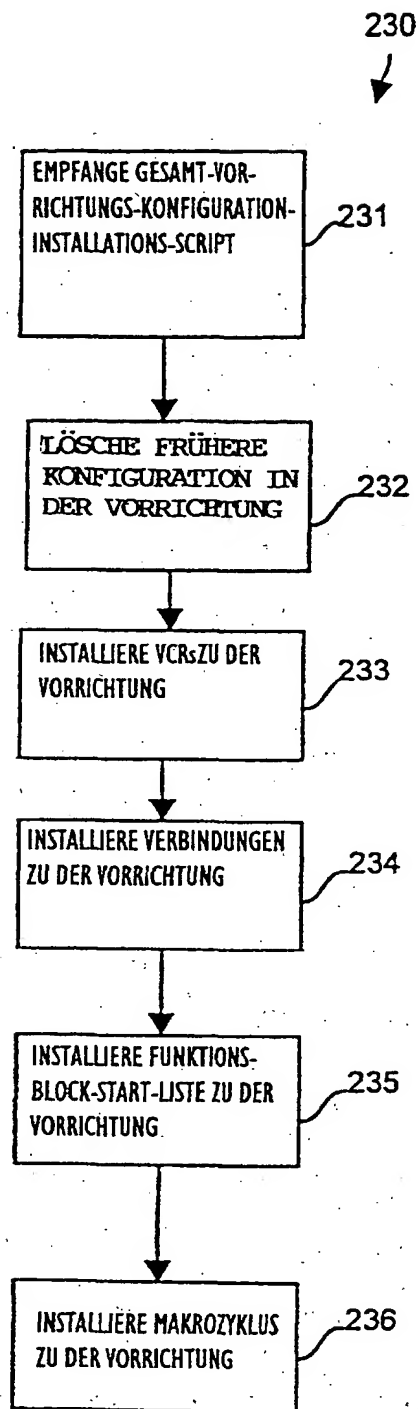


FIG. 9

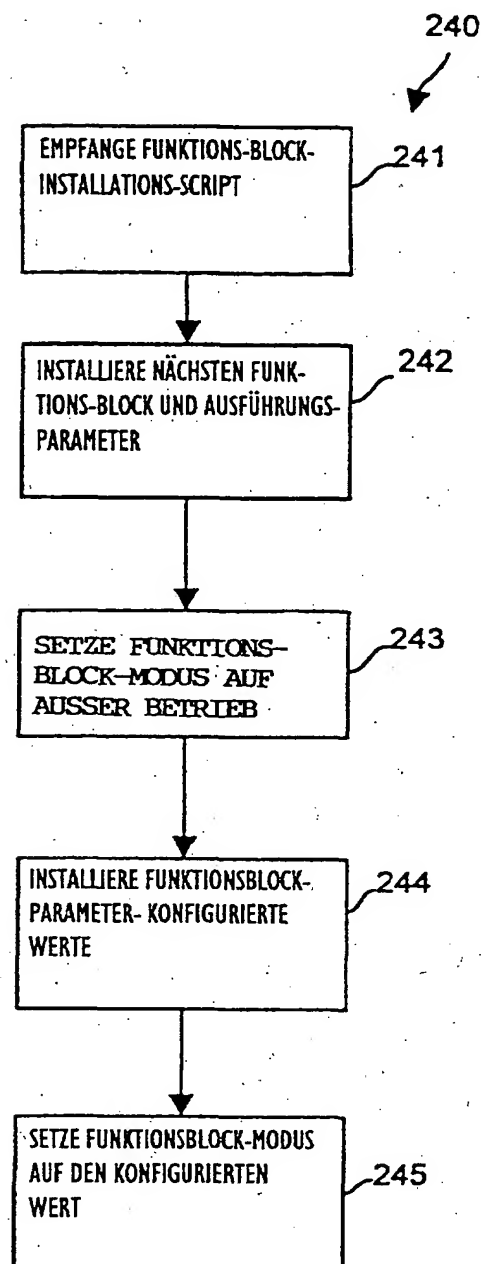


FIG. 10

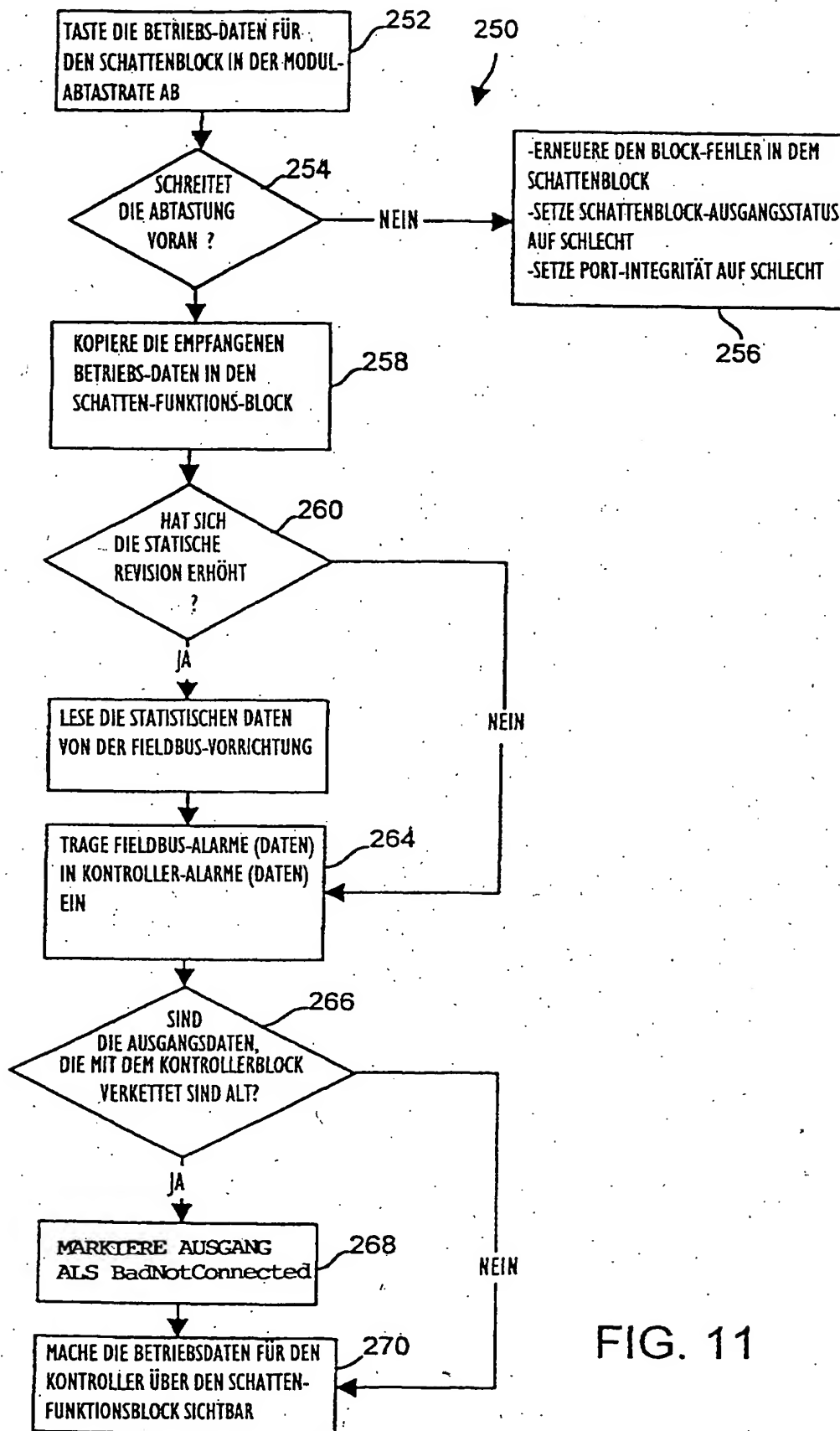


FIG. 11

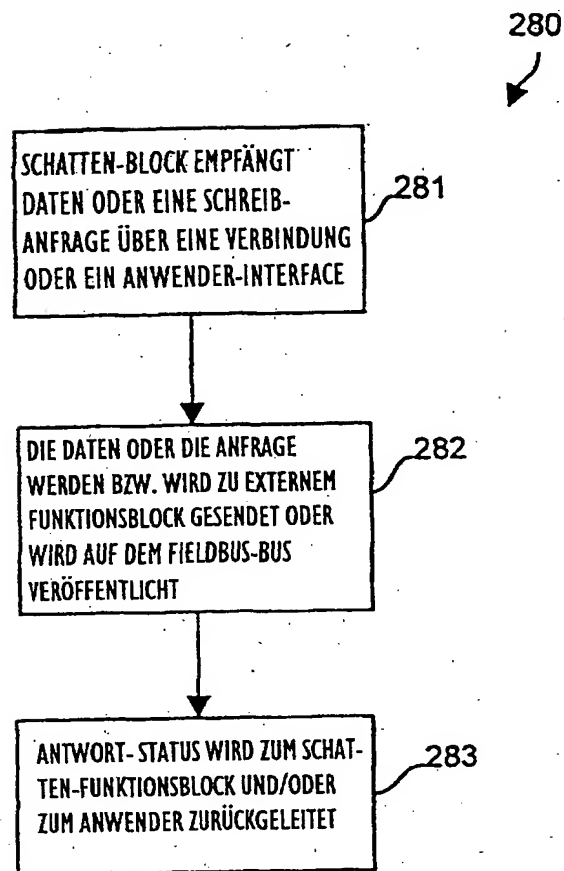


FIG. 12